



GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO FINAL DE GRADO

---

**Seguimiento de contratos de obligación  
de FOBESA  
mediante metodologías ágiles**

---

*Autor:*

Josep Lluís FERRER PRATS

*Supervisor:*

Javier MUÑOZ FERRARA

*Tutora académica:*

Lledó MUSEROS CABEDO

Fecha de lectura: 22 de Junio de 2018

Curso académico 2017/2018

## Resumen

Este documento describe el proceso de desarrollo de la aplicación web, para clientes PC, tabletas y móviles, de seguimiento de los contratos con los clientes de la empresa FOBESA. A partir de este punto el proyecto se denominará a todos los efectos 'Informe de producción de FOBESA'. Este es el nombre por el que se conoce en ADC Infraestructuras y Sistemas dicho informe. Partiendo de la necesidad de construir la aplicación, se ha realizado el análisis, diseño, implementación y pruebas utilizando una metodología iterativa e incremental (*Scrum*), obteniendo como resultado una aplicación web completamente funcional.

El desarrollo de la aplicación se ha realizado durante la estancia en prácticas en la empresa ADC Infraestructuras y Sistemas, empresa integrada en el Grupo Gimeno.

La aplicación conjuntamente con este documento conforman el Trabajo de final de grado (TFG) en el Grado en ingeniería informática de la Universitat Jaume I.

## Palabras clave

*Scrum, Spring boot, Angular, Git*

## Keywords

*Scrum, Spring boot, Angular, Git*

# Índice general

<b>1. Introducción</b>	<b>11</b>
1.1. Contexto del proyecto . . . . .	11
1.2. Motivación . . . . .	13
1.3. Objetivos del proyecto . . . . .	15
1.4. Estructura de la memoria . . . . .	15
<b>2. Descripción del proyecto</b>	<b>17</b>
2.1. Descripción general . . . . .	17
2.2. Descripción funcional . . . . .	17
2.3. Alcance . . . . .	18
2.4. Tecnologías y herramientas . . . . .	19
<b>3. Planificación del proyecto</b>	<b>23</b>
3.1. Metodología . . . . .	23
3.1.1. ¿ Por qué utilizar una metodología ágil en este proyecto? . . . . .	24
3.1.2. Especificación de la metodología . . . . .	24
3.1.3. Historias de usuario . . . . .	25
3.1.4. Puntos de historia . . . . .	25
3.1.5. Pila del producto ( <i>Product backlog</i> ) . . . . .	26

3.2.	Planificación . . . . .	26
3.3.	Estimación de recursos y costes del proyecto . . . . .	27
3.4.	Seguimiento del proyecto . . . . .	30
<b>4.</b>	<b>Análisis y diseño del sistema</b>	<b>31</b>
4.1.	Análisis del sistema . . . . .	31
4.1.1.	Pila de producto inicial ( <i>initial product backlog</i> ) . . . . .	31
4.1.2.	Pila de producto final . . . . .	33
4.2.	Diseño del sistema . . . . .	34
4.2.1.	Diseño de datos . . . . .	34
4.2.2.	Diseño de la arquitectura . . . . .	34
4.2.2.1.	Estilo arquitectónico . . . . .	35
4.2.2.2.	Arquitectura del sistema . . . . .	35
4.2.2.3.	Arquitectura de la aplicación web . . . . .	36
4.2.3.	Diseño de la interfaz de usuario (GUI) . . . . .	36
4.2.3.1.	Interfaz gráfica . . . . .	37
<b>5.</b>	<b>Descripción de la implementación y pruebas</b>	<b>45</b>
5.1.	Lenguajes y herramientas utilizado en la implementación . . . . .	45
5.2.	Proyectos y módulos de software . . . . .	46
5.3.	Patrones de diseño . . . . .	48
5.3.1.	Patrón Singleton y su variante Multiton . . . . .	48
5.3.2.	Patrón Builder . . . . .	49
5.3.3.	Patrón Dependency injection . . . . .	50
5.3.4.	Patrón Observer . . . . .	51
5.4.	Pruebas unitarias . . . . .	52

5.5.	Pruebas de aceptación . . . . .	52
5.6.	Aseguramiento de la calidad del código . . . . .	53
<b>6.</b>	<b>Construcción e Iteraciones</b>	<b>57</b>
6.1.	<i>Sprint</i> 1 . . . . .	57
6.1.1.	Planificación . . . . .	57
6.1.1.1.	Tareas planificadas . . . . .	57
6.1.2.	Pruebas de aceptación . . . . .	58
6.1.3.	Pruebas unitarias . . . . .	59
6.1.4.	Resultados obtenidos . . . . .	61
6.2.	<i>Sprint</i> 2 . . . . .	61
6.2.1.	Planificación . . . . .	62
6.2.1.1.	Actualización de la pila del producto . . . . .	63
6.2.1.2.	Tareas planificadas . . . . .	64
6.2.2.	Pruebas de aceptación . . . . .	65
6.2.3.	Pruebas unitarias . . . . .	65
6.2.4.	Resultados obtenidos . . . . .	66
6.3.	<i>Sprint</i> 3 . . . . .	71
6.3.1.	Planificación . . . . .	71
6.3.1.1.	Actualización de la pila del producto . . . . .	71
6.3.1.2.	Tareas Planificadas . . . . .	72
6.3.2.	Pruebas de aceptación . . . . .	73
6.3.3.	Pruebas unitarias . . . . .	75
6.3.4.	Resultados obtenidos . . . . .	76
6.4.	<i>Sprint</i> 4 . . . . .	76

6.4.1. Planificación . . . . .	76
6.4.1.1. Actualización de la pila del producto . . . . .	77
6.4.1.2. Tareas planificadas . . . . .	80
6.4.2. Pruebas de aceptación . . . . .	80
6.4.3. Pruebas unitarias . . . . .	81
6.4.4. Resultados obtenidos . . . . .	81
6.5. <i>Sprint</i> 5 . . . . .	82
6.5.1. Planificación . . . . .	82
6.5.1.1. Actualización de la pila del producto . . . . .	82
6.5.1.2. Tareas planificadas . . . . .	83
6.5.2. Pruebas de aceptación . . . . .	84
6.5.3. Pruebas unitarias . . . . .	85
6.5.4. Resultados obtenidos . . . . .	86
<b>7. Conclusiones</b>	<b>91</b>
7.1. Modificaciones y futuras extensiones . . . . .	91
<b>Bibliografía</b>	<b>91</b>
<b>A. Descripción de la metodología <i>Scrum</i> para proyectos de software</b>	<b>95</b>
A.1. <i>Sprints</i> . . . . .	95
A.2. Requisitos funcionales . . . . .	96
A.3. Reuniones . . . . .	96
<b>B. Datos en formato JSON para los tests de aceptación</b>	<b>99</b>
<b>C. Sentencias de la base de datos H2 para tests</b>	<b>109</b>

# Índice de figuras

1.1. Pantalla de solicitud del Informe de producción existente a sustituir. . . . .	13
1.2. Resultado de ejemplo del Informe de producción existente a sustituir (los datos se han manipulado siguiendo las normas de confidencialidad del grupo Gimeno). . . . .	14
2.1. Ilustración de flujo de modificaciones con Git Flow. . . . .	20
2.2. Ejemplo de pantalla de Gitlab con ramas master, develop y una rama para nueva funcionalidad. . . . .	21
2.3. Ejemplo de pantalla de solicitud de merge en Gitlab. . . . .	22
3.1. Tarjeta con la estructura de Historia de usuario. . . . .	25
3.2. Planificación temporal inicial del proyecto y las dependencias entre tareas junto con la asignación inicial de historias de usuario del primer Sprint . . . . .	28
3.3. Coste económico del proyecto. . . . .	29
3.4. Diagrama de Gantt con la temporización del proyecto. . . . .	30
4.1. Diagrama de casos de uso del proyecto (a título ilustrativo). . . . .	32
4.2. Pila de producto ( <i>Product Backlog</i> ) inicial. . . . .	33
4.3. Diagrama de dependencias de las clases del modelo del proyecto. . . . .	35
4.4. Diagrama de clases con atributos del modelo del proyecto. . . . .	38
4.5. Arquitectura software del sistema en desarrollo. . . . .	39
4.6. Arquitectura software del sistema en ejecución. . . . .	39
4.7. Prototipo de la pantalla de detalle del Contrato de obligación. . . . .	40

4.8. Prototipo de la pantalla de obligaciones del Contrato de obligación. . . . .	41
4.9. Prototipo de la pantalla de actuaciones del Contrato de obligación. . . . .	42
4.10. Prototipo de la pantalla de gestión económica del Contrato de obligación. . . . .	43
5.1. Ejemplo de pantalla de despliegue de la aplicación en Jenkins. . . . .	48
5.2. Patrón Singleton. . . . .	49
5.3. Patrón Multiton. . . . .	49
5.4. Patrón Builder. . . . .	50
5.5. Patrón Inversión de dependencias. . . . .	51
5.6. Patrón Observer. . . . .	52
5.7. Inspección de código propio de la organización. . . . .	54
5.8. Metricas de analisis estatico de IntelliJ. . . . .	55
6.1. Estado de las tareas asignadas al <i>Sprint</i> 1 al finalizar dicho <i>Sprint</i> . . . . .	62
6.2. <i>Incremento</i> del producto tras <i>Sprint</i> 1. . . . .	63
6.3. Product backlog tras planificar el <i>Sprint</i> 2. . . . .	66
6.4. Estado de las tareas asignadas al <i>Sprint</i> 2 al finalizar dicho <i>Sprint</i> . . . . .	67
6.5. <i>Incremento</i> del producto tras <i>Sprint</i> 2 (Región superior). . . . .	68
6.6. <i>Incremento</i> del producto tras <i>Sprint</i> 2 (Región intermedia). . . . .	69
6.7. <i>Incremento</i> del producto tras <i>Sprint</i> 2 (Región inferior). . . . .	70
6.8. Product backlog tras planificar el <i>Sprint</i> 3. . . . .	73
6.9. Estado de las tareas asignadas al <i>Sprint</i> 3 al finalizar dicho <i>Sprint</i> . . . . .	77
6.10. <i>Incremento</i> del producto tras <i>Sprint</i> 3 (Región superior). . . . .	78
6.11. <i>Incremento</i> del producto tras <i>Sprint</i> 3 (Región inferior). . . . .	79
6.12. Product backlog tras planificar el <i>Sprint</i> 4. . . . .	81
6.13. Estado de las tareas asignadas al <i>Sprint</i> 4 al finalizar dicho <i>Sprint</i> . . . . .	87



6.14. <i>Incremento</i> del producto tras <i>Sprint</i> 4. . . . .	88
6.15. Product backlog tras planificar el <i>Sprint</i> 5. . . . .	88
6.16. Estado de las tareas asignadas al <i>Sprint</i> 5 al finalizar dicho <i>Sprint</i> . . . . .	89
6.17. <i>Incremento</i> del producto tras el <i>Sprint</i> 5. . . . .	90



# Capítulo 1

## Introducción

En este capítulo se introduce la empresa, la motivación del proyecto, los objetivos del mismo y la estructura de esta memoria.

### 1.1. Contexto del proyecto

El proyecto cuya propuesta se presenta en este documento, se ha desarrollado en la empresa ADC Infraestructuras y sistemas, integrada en el Grupo Gimeno. El proyecto se desarrolla para el cliente FOBESA. A continuación identificamos la empresa de forma genérica.

El Grupo Gimeno, como se puede leer en la página web *Grupo Gimeno* [1], se auto-define como:

Se funda en Castellón con la constitución de FACSA, empresa pionera en la gestión del ciclo integral del agua, hace más de 140 años. Desde entonces Grupo Gimeno ha evolucionado de forma paralela a la sociedad y ha apostado por la innovación y la orientación hacia nuevos y decisivos sectores económicos.

Fruto de esta evolución, surgen tres divisiones: Gimeno Servicios, Gimeno Construcción y Gimeno Turismo y Ocio.

Estos esfuerzos y visión innovadora se han traducido en más de 30 empresas que operan en todo el territorio nacional.

En Grupo Gimeno trabajamos para dar servicio a más de un millón de personas gracias al esfuerzo de los más de 4.100 profesionales que forman parte de nuestro equipo humano.

Con una facturación agregada de 240 millones de euros, centramos nuestra capacidad en evolucionar y encarar el futuro con el desafío de responder a los nuevos retos de crecimiento.

Extraído de la página web *Grupo Gimeno* [1], la empresa ADC Infraestructuras y sistemas, ese presenta como:

Empresa especializada en el desarrollo de los sistemas de telecontrol del ciclo integral del agua. ADC crea sus propias aplicaciones informáticas para la gestión de abastecimientos y el control de estaciones depuradoras.

En concreto su Área de tecnologías de la información (página web *Grupo Gimeno* [1]):

Dentro de las áreas de actividad de la empresa ADC se encuentra el Servicio de Informática y Comunicaciones, que se divide en tres áreas: software, datacenter y comunicaciones y micro-informática. El área software principalmente desarrolla aplicaciones a medida para la gestión de las empresas del grupo. También realiza un servicio de consultoría de modelización de datos y ofrece asesoramiento y mantenimiento en la adquisición de aplicaciones de terceros. De entre todos los servicios ofertados destacamos: Desarrollo a medida y mantenimiento de software de gestión Atención al usuario. En todas nuestras aplicaciones proporcionamos atención al usuario de forma permanente, cursos de formación y soporte funcional Consultoría de modelización de datos: Instalación y actualización de bases de datos Administración de bases de datos Desarrollo de procesos ETL (extracción, transformación y carga de datos) Servicios de intranet/extranet. Gestionamos la contratación de proyectos Web. Los proyectos / servicios de estas soluciones pueden ser: Herramientas software para comercio electrónico Páginas Web corporativas Desarrollo de intranets Consultoría y soporte en la adquisición de herramientas software de terceros. Proporcionamos asesoramiento en la adquisición de herramientas software no desarrolladas por nuestra empresa disponibles en el mercado. Además, ofrecemos el seguimiento en la implantación, mantenimiento y soporte funcional de estas aplicaciones

Mientras que la autodescripción de FOBESA es:

Fobesa nace en la provincia de Castellón en la década de los 60, pero con el tiempo se ha expandido geográficamente y ha diversificado su actividad. Desde sus inicios, como empresa centrada en el abastecimiento de aguas potables en la localidad de Benicàssim, Fobesa se ha convertido en un referente en la gestión y la protección del medio ambiente.

Nuestro compromiso con la protección del medio ambiente se plasma en la introducción de mejoras constantes en los servicios que ofrecemos. Destinamos importantes recursos a la investigación y a la aplicación de las nuevas tecnologías al sector medioambiental y contamos con un equipo humano profesional, responsable y especializado.

La especialización nos ha permitido crecer más allá de la provincia de Castellón y en la última década hemos multiplicado nuestros servicios y el número de poblaciones en las que estamos presentes. Fobesa desarrolla su actividad en unas 50 poblaciones de las provincias de Castellón, Teruel, Alicante y Albacete, donde atiende cada día a cerca de 800.000 personas.

Fobesa forma parte de Gimeno Servicios, que también integra a otras empresas: Facsa, Iproma, Sitra-Prodesa Agua y Gimeno Marítimo.

Concretando el contexto, el proyecto se ha desarrollado en la empresa ADC Infraestructuras y Sistemas. En este caso se ha analizado, diseñado e implementado un producto software para la empresa FOBESA, una de las empresas del Grupo Gimeno, cliente habitual de ADC Infraestructuras y Sistemas.

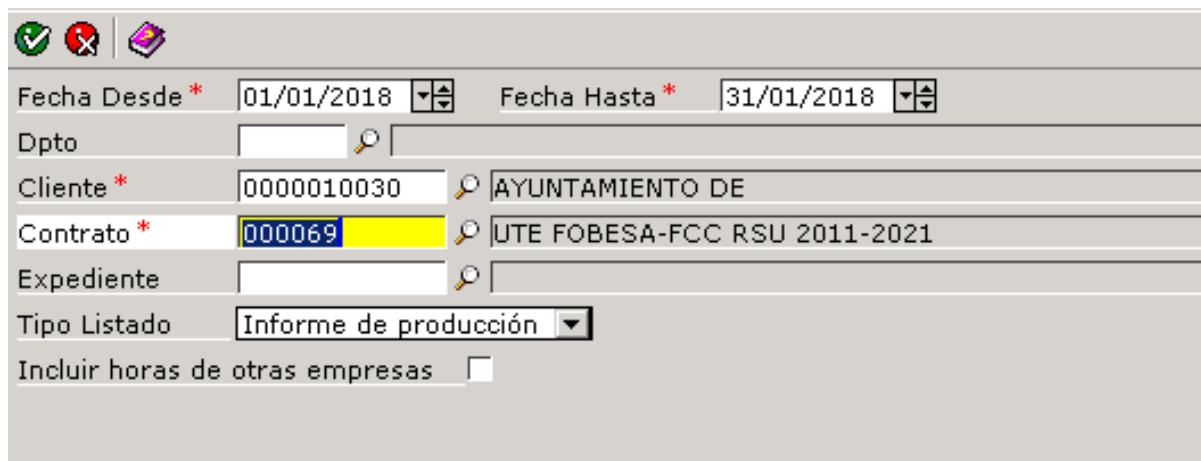
## 1.2. Motivación

El proyecto se centra en los Contratos de obligación. Contrato de obligación es el documento que firma la empresa con un cliente final, que puede ser Ayuntamiento, Diputación, Comunidad autónoma,... El objeto del contrato es la prestación de un servicio , habitualmente de limpieza o recogida de basuras.

La motivación del proyecto surge del hecho de que en este momento se genera una hoja de cálculo, denominada Informe de Producción, para hacer el seguimiento del Contrato de obligación. Ésta se produce con consultas de información realizadas con VisualBasic a la base de datos. En esta información consta:

- la información estática del contrato;
- la información de parámetros de cumplimiento del contrato (horas dedicadas, trabajadores asignados al contrato, medios de transporte,...); la información de pesajes de residuos producidos como resultado del contrato;
- los costes internos que repercute este contrato. Este apartado es solo para uso interno de la empresa. No debe ser visible al cliente final.

En las figuras 1.1 y 1.2 se puede ver un ejemplo de la hoja de cálculo actual a sustituir (los datos se han manipulado siguiendo las normas de confidencialidad del del Grupo Gimeno)



Fecha Desde *	01/01/2018	Fecha Hasta *	31/01/2018
Dpto			
Cliente *	0000010030		AYUNTAMIENTO DE
Contrato *	000069		UTE FOBESA-FCC RSU 2011-2021
Expediente			
Tipo Listado	Informe de producción		
Incluir horas de otras empresas	<input type="checkbox"/>		

Figura 1.1: Pantalla de solicitud del Informe de producción existente a sustituir.

Esta forma de trabajar provoca varios inconvenientes:

- El usuario debe estar dentro de la intranet de la empresa para hacer uso de esta utilidad.
- El usuario de la aplicación no puede ser el cliente final porque no existe filtrado de información para que el cliente solo vea sus datos y porque el usuario debe estar ubicado en la intranet.

	A	B	C	D	E	F
1	<b>1. Datos origen del servicio</b>					
2	Fecha inicio	Fecha fin	Fecha proroga	Importe adjudicaci	Revisiones económicas	
3	01/02/2011	01/02/2021				
4						
5						
6	<b>1.1 Mejoras de contrato</b>					
7	Descripción	Total €/Año	Total €/Contrato	Coste Real	Puntuación	Realizada
8						
9						
10	<b>2. Contratos Obligaciones</b>					
11	<b>Nules LV</b>					
12	Recursos	H. Programadas	H. Realizadas	Total dif. horas		
13	Barredora 3m3	177,14		99999		
14	Barredora 5m3	88,57		100000		
15	Camión cisterna	0		100001		
16	Camión plataforma	0		100002		
17	Camión Trasero 14-16 m3	11,07		100003		
18	Camión Trasero 18-20 m3	0		100004		
19	Conductor día	11,07		100005		
20	Encargado LV	177,23		100006		
21	Hidrolimpiadora	14,75		100007		
22	Peón día	1247,13		100008		
23	Piaggio	177,14		100009		
24	Vehículo encargado	177,23		100010		
25						
26						
27	<b>3. Pesajes</b>					
28	Tipo Residuo	Cantidad	Unidad			
29						
30						
31	<b>4. Contenedores</b>					
32	Col 01	Col 02	Col 03	Col 04	Col 05	
33						
34	<b>5. Gestión Ecoparque</b>					
35	Nº usuarios					
36						
37	<b>6. Vaciado de Contenedores</b>					
38	Residuo	Fecha	Nº Contenedores levantad	Peso		
39						
40	<b>7. Gestión económica del servicio</b>					
41	Tipo Coste	Descripción	Importe			
42		99301 Generales Fobesa	99999			
43		99302 Generales Area Castellon	100000			
44		99308 Generales Area Castellon Taller	100001			

Figura 1.2: Resultado de ejemplo del Informe de producción existente a sustituir (los datos se han manipulado siguiendo las normas de confidencialidad del grupo Gimeno).

- La aplicación solo funciona en un ordenador con Excel y que pueda ejecutar VisualBasic.
- El mantenimiento de la aplicación es diferente al resto de aplicaciones de la empresa.

La empresa desea sustituir esta hoja de cálculo por una aplicación web y móvil. Esto tendrá diversas ventajas:

- El usuario podrá estar ubicado en cualquier localización.
- El usuario de la aplicación podrá ser el cliente final.
- La aplicación funcionará en cualquier ordenador con un navegador o en un móvil.
- El mantenimiento de la aplicación se realizará de la misma forma que las otras aplicaciones de la empresa.

Después de completar el proyecto los beneficios para la empresa serán:

- El mantenimiento de la aplicación será estándar.

- La aplicación podrá evolucionar y crecer junto con las aplicaciones del grupo.
- La empresa tendrá la ventaja competitiva de cara al cliente final de poder ofrecerle la consulta de la información correspondiente al cumplimiento de su contrato.
- Los usuarios internos de la empresa tendrán una mayor flexibilidad respecto a dispositivos para consultar estos datos.

### **1.3. Objetivos del proyecto**

El principal objetivo de este proyecto es ofrecer el informe de producción de Fobesa tanto a los usuarios internos como externos de la empresa. El objetivo principal se puede desglosar en los siguientes subobjetivos:

- Crear una aplicación web y móvil que permitirá a los usuarios internos consultar el informe en cualquier dispositivo que disponga de un navegador, tanto dentro como fuera de la intranet.
- Crear una aplicación web y móvil que permitirá a los clientes finales consultar en informe, limitado a los datos a los que tienen autorización.
- Permitir un mantenimiento estándar de la aplicación siguiendo la forma de trabajar de la empresa.

Al final del proyecto se espera alcanzar el objetivo de ofrecer la aplicación web a los usuarios internos de la empresa. Dejando para una futura ampliación la inclusión de clientes finales como usuarios de la aplicación.

### **1.4. Estructura de la memoria**

En el capítulo 2 se hace una descripción pormenorizada del proyecto a desarrollar. El capítulo 3 muestra la metodología, planificación y seguimiento que se ha realizado del proyecto. En el capítulo 4 se especifica el análisis y se hace un diseño de todos los aspectos de la aplicación. En el capítulo 5 se describe la fase de construcción y tests del proyecto. Finalmente en el capítulo 6 se detallan la iteraciones para construir el producto final.





## Capítulo 2

# Descripción del proyecto

### 2.1. Descripción general

La empresa ADC Infraestructuras y servicios ha decidido desarrollar el proyecto Informe de producción dentro de la estructura de su portal de aplicaciones. La aplicación, al igual que el resto de aplicaciones de la empresa constará de las siguientes capas:

- servicios REST (*Representational State Transfer*) que provee a la aplicación de la información extraída de los distintos orígenes de información siguiendo la filosofía REST:
  - Servicio de Contratos de obligación. Accede a la base de datos SQLServer con los datos actualizados de los contratos de obligación.
  - Servicio de Partes de trabajo, tanto de empleados como de vehículos
  - Servicio con la información histórica de sensores con pesos de contenido de camiones. Este servicio es ajeno al proyecto y ha sido desarrollado por la empresa del grupo IOTSens.
- servicio REST que accede a la anterior capa de servicios. Esta capa prepara la información tal como se quiere visualizar. Es una capa que combina reglas de negocio con presentación de la vista.
- Vista. Accede a la anterior capa de servicios para generar los componentes de la Vista a partir de la información recibida.

### 2.2. Descripción funcional

Se trata de una aplicación web para consultar el informe de producción de Fobesa. Esta debe permitir consultar:

- Los contratos de obligación.

- Las mejoras de los contratos de obligación.
- Los recursos y sus horas programadas y realizadas.
- Los pesajes de residuos por tipo de residuo.
- El vaciado de contenedores por fecha.
- La gestión económica del servicio.

En el futuro a la aplicación accederán dos roles:

- Usuarios internos de la empresa. Podrán acceder a toda la información de la aplicación desde cualquier dispositivo (PC, tableta o móvil).
- Usuarios de la empresa con la que se ha firmado el contrato de obligación (Ayuntamientos, Diputaciones, Entidades autonómicas, ...). Podrán acceder a toda la información de la aplicación excepto a la gestión económica del servicio, desde cualquier dispositivo (PC, tableta o móvil).

## 2.3. Alcance

Este proyecto consiste en desarrollar una aplicación web para consultar el informe de producción de Fobesa. Concretando el alcance, debe permitir consultar:

- Los contratos de obligación.
- Las mejoras de los contratos de obligación.
- Los recursos y sus horas programadas y realizadas.
- Los pesajes de residuos por tipo de residuo.
- El vaciado de contenedores por fecha.
- La gestión económica del servicio.

Se excluye del alcance del proyecto la construcción de la base de datos y los diversos servicios de captura de datos a ser consultados. La base de datos ya esta construida y mantenida. Los servicios a consultar están contruidos o acabándose de construir.

También se excluye del alcance la integración en el portal del Grupo y la gestión de roles (que forma parte de una futura mejora). A nivel de roles el alcance incluye la consideración de un solo rol, que es el usuario interno de la empresa sin restricciones de acceso a partes de la información.

## 2.4. Tecnologías y herramientas

Las tecnologías han sido consensuadas con la empresa de prácticas y son las siguientes:

- El acceso a las bases de datos se realizarán utilizando JPA (*Java Persistence Api*) y QueryDSL.
- Como backend se construirán servicios REST. El lenguaje usado será Java.
- El frontend de la aplicación se construirá con Angular. La lógica de visualización se escribirá en Typescript compilado a JavaScript, mientras la capa de acceso al servicio REST tendrá la lógica de la aplicación y visualización escrita en Java.

Como herramientas principales a utilizar:

- IntelliJ como herramienta de desarrollo de la aplicación.
- JUnit como framework de desarrollo de pruebas unitarias y de integración.
- GitLab como aplicación de gestión de versiones.
- Jenkins como herramienta de integración continua.

También se han consensuado el protocolo ***Git Flow*** para control de versionado. Los principales pasos del ciclo habitual de desarrollo con *Git Flow* son:

1. El desarrollador crea una rama específica para la nueva funcionalidad a desarrollar a partir de la rama Develop.
2. Una vez probada la nueva funcionalidad se genera un *Request merge* a la rama develop asignado al Supervisor del proyecto.
3. El Supervisor valida o no los cambios realizados.
4. Si el supervisor valida los cambios realizados realiza un Merge con la rama Develop. Con ello Develop ya tiene la nueva funcionalidad.
5. Si el supervisor no valida los cambios realizados anota en GitLab los cambios a realizar.

En la figura 2.1 se puede ver los pasos completos, extraídos de la página web [4].

En la figura 2.2 se puede ver las ramas master, develop y una rama creada para añadir una funcionalidad en Gitlab.

En la figura 2.3 se puede ver una pantalla de solicitud de merge en Gitlab.

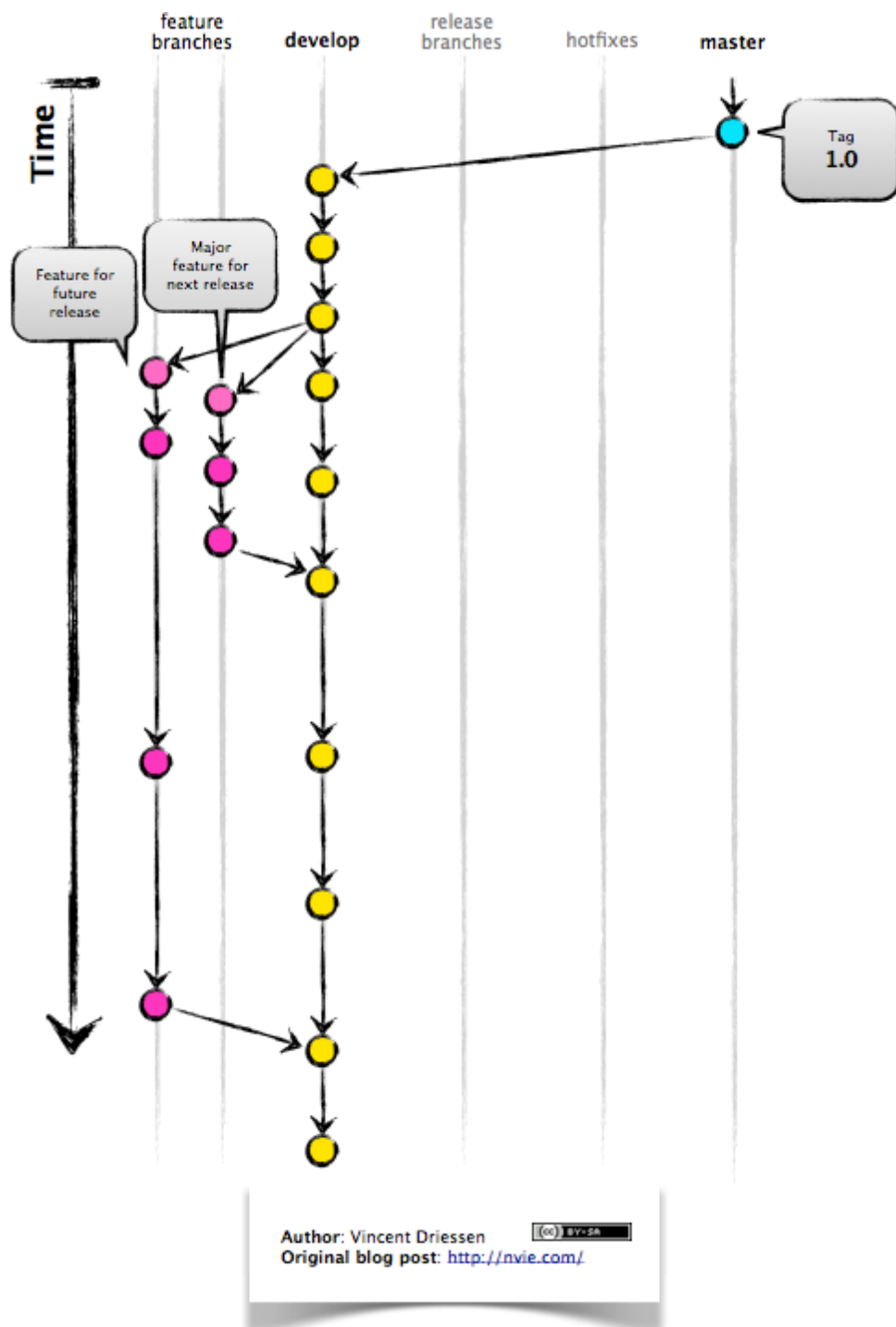


Figura 2.1: Ilustración de flujo de modificaciones con Git Flow.

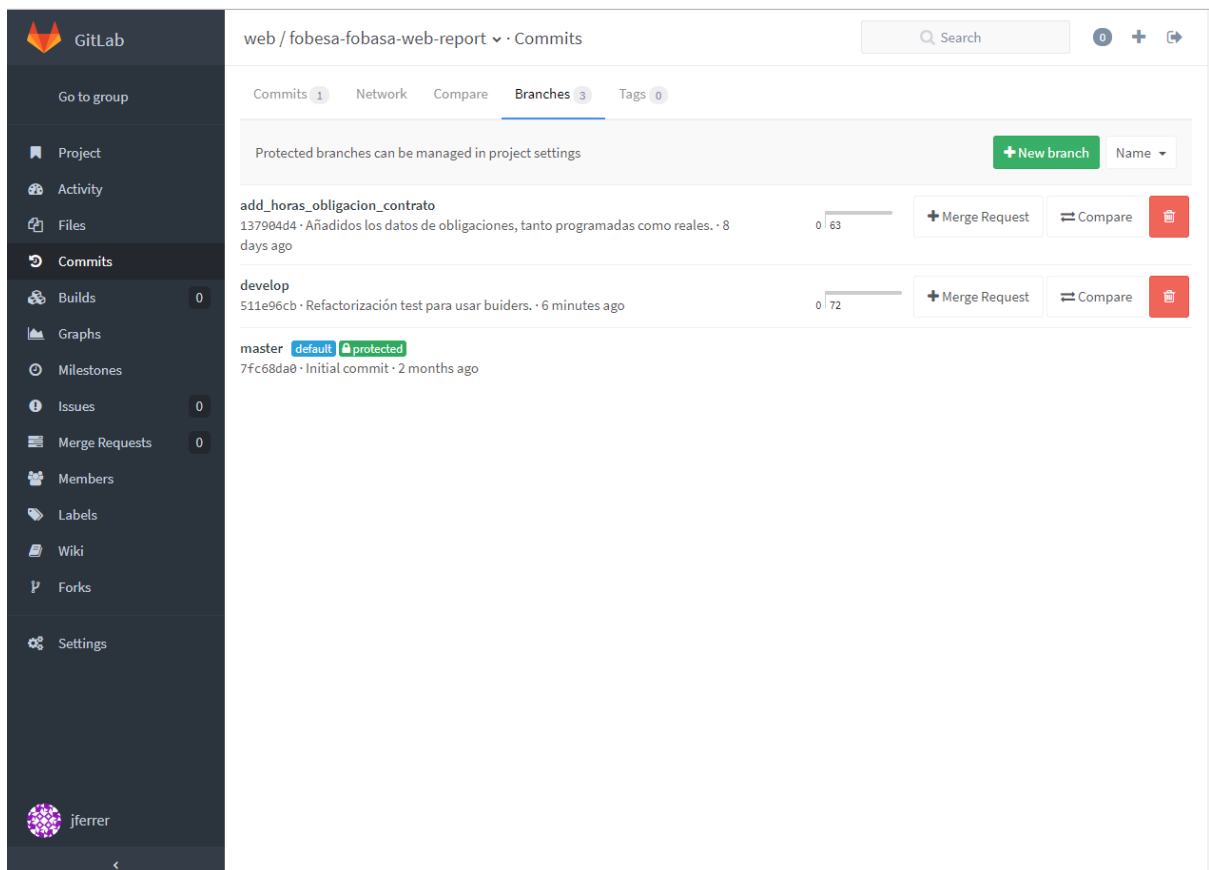


Figura 2.2: Ejemplo de pantalla de Gitlab con ramas master, develop y una rama para nueva funcionalidad.

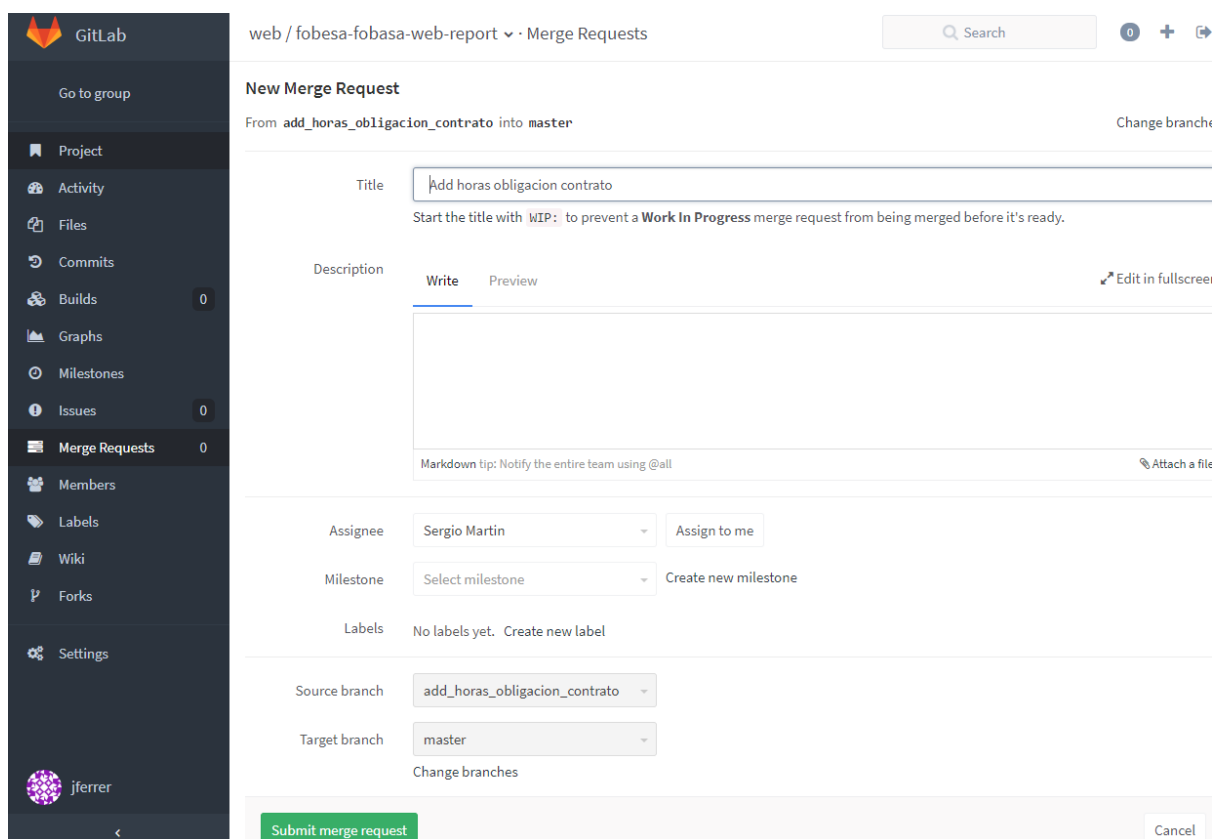


Figura 2.3: Ejemplo de pantalla de solicitud de merge en Gitlab.

## Capítulo 3

# Planificación del proyecto

### 3.1. Metodología

Según se desprende de (*Mike Cohn. Succeeding with agile*) [12], más de la mitad de los problemas que aparecen en desarrollo de los proyectos tienen su origen en los requisitos originados a partir de:

- Mala especificación de los requisitos.
- Cambios de los requisitos funcionales.
- Poca implicación de todas las partes del proyecto.
- Escasa comunicación entre el cliente y el equipo de desarrollo.

En las metodologías de desarrollo tradicionales se intenta definir exhaustivamente los requisitos y funcionalidad del sistema a desarrollar. El cliente no ve el producto hasta la última fase de desarrollo. Cualquier ajuste sobre requerimientos en esta última fase implica un gran coste de modificación. Esta es la principal razón por lo que estas metodologías tradicionales solo son aconsejables en proyectos con especificaciones muy cerradas e inmutables, con una gran implicación del cliente en la fase de análisis del proyecto y un elevado compromiso de no modificar lo establecido.

Las ingenierías clásicas (arquitectura, industrial, mecánica, eléctrica, etc.) necesitan utilizar este tipo de ciclos de vida en cascada o predictivos. Dependen de un diseño inicial, exhaustivo e inmutable. Una vez construidos los cimientos difícilmente podrá rediseñarse el plano de un edificio.

El desarrollo de software es diferente a este tipo de proyectos, ya que se pueden realizar ciertas modificaciones siguiendo una construcción del producto en iteraciones bien diseñadas.

### 3.1.1. ¿ Por qué utilizar una metodología ágil en este proyecto?

La razón principal es no tener una lista cerrada y bien definida de requerimientos del proyecto. Disponemos de un ejemplo de la pantalla Excel a sustituir (figura 1.2) y parte del código SQL (*Structured Query Language*) y VisualBasic que genera la hoja mencionada. Pero no tenemos todo el código Visualbasic ni todas las sentencias SQL.

Además, los requerimientos no los consideramos cerrados. La aplicación web a desarrollar proporciona más opciones de presentación que la hoja Excel tiene implementadas. Podemos presentar información en forma de gráficas o manejar autorizaciones para diferentes tipos de usuarios. Y todas estas mejoras no han sido establecidas a priori.

El uso de estas metodologías nos aportara varios beneficios:

- Obtener un PMV (Producto mínimo viable) desde las primeras fases de desarrollo.
- *Feedback* del cliente en cada iteración de la fase de desarrollo.
- Gestionar cambios de forma flexible.
- Producción incremental del producto (cada *Sprint* añade valor añadido al producto).
- Reducción de riesgos a largo plazo.
- Gestionar la complejidad del proyecto.
- Conocer el progreso real del proyecto en todo momento.

### 3.1.2. Especificación de la metodología

La metodología a utilizar en el desarrollo del proyecto ha sido consensuada con la empresa de prácticas. Se ha decidido usar una metodología ágil, en concreto *Scrum*, para la evolución del proyecto. La metodología *Scrum* está detallada en el anexo A.

Concretamente se ha decidido esta metodología por las siguientes características:

- Estabilidad muy baja de los requisitos. Como ya hemos mencionado los requisitos no están cerrados ni son inmutables
- Tamaño del proyecto. Se trata de un proyecto de 300 horas, que se considera de tamaño mediano.
- Condiciones favorable de la organización. La empresa trabaja habitualmente con este tipo de metodología.
- Entorno de desarrollo en continua evolución. En el mundo del desarrollo de software los cambios son continuos.



### 3.1.3. Historias de usuario

En las metodologías ágiles la especificación de las necesidades se realiza mediante historias de usuario que son, básicamente, lo que el cliente quiere que cumpla la aplicación a implementar.; es decir, son una descripción breve de la funcionalidad software tal y cómo el usuario la percibe (*M. Cohn, Users histories applied*) [11].

Las historias de usuario se escriben en tarjetas que contienen la información de a figura 3.1 extraída de la página (*Historias de usuario*) [2]

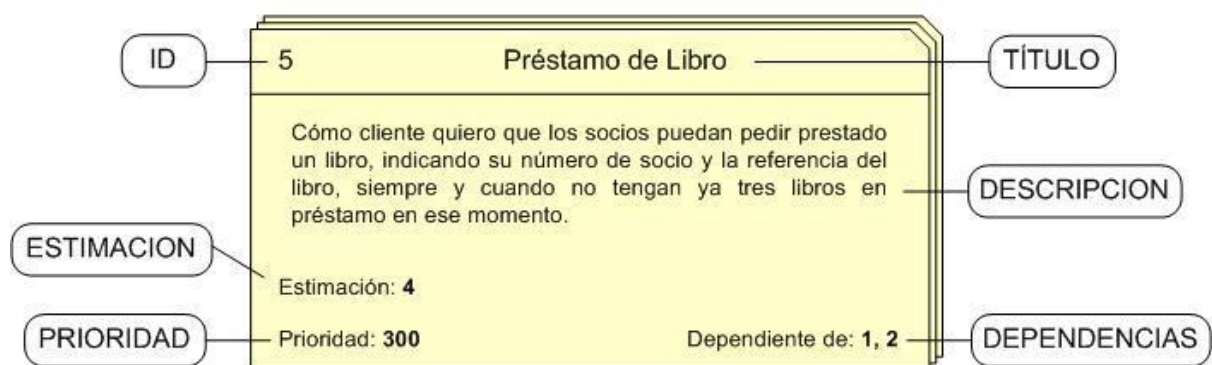


Figura 3.1: Tarjeta con la estructura de Historia de usuario.

- **Id:** identifica la historia de usuario.
- **Título:** describe brevemente la historia de usuario.
- **Descripción:** debe responder a las tres preguntas: ¿Quien?, ¿Qué? y ¿Para qué?.  
**Como**[rol de usuario]**Necesito**[objetivo]**Para**[beneficio]
- **Estimación:** cálculo del tiempo necesario de desarrollo de la historia de usuario en unidades de desarrollo o **puntos de historia** (unidades de tiempo teórico de desarrollo/persona).
- **Prioridad:** Valor numérico o discreto que indica la importancia para el cliente de esta funcionalidad.

### 3.1.4. Puntos de historia

Las metodologías ágiles han cambiado muchos de los conceptos del desarrollo de software. Uno de ellos es la forma de estimar el esfuerzo de desarrollo. En las historias de usuario se valora la complejidad de cada una de ellas respecto a otras. Posteriormente cada organización aplica una traducción temporal diferente al punto de historia.

En nuestro proyecto **un punto historia de usuario equivale aproximadamente a 1 hora de trabajo**. Hemos optado por una unidad de tiempo mínima que no necesite ser dividida.

Consideramos que nunca se valorará una historia de usuario, por pequeña que sea, en menos de 1 hora.

Para la realización de este proyecto disponemos de **5 horas diarias de trabajo**. Descontadas las reuniones, descansos e interrupciones habituales, consideramos que **cada día tendremos 4 horas de desarrollo de historias de usuario**.

### 3.1.5. Pila del producto (*Product backlog*)

Una vez descritos los conceptos **historia de usuario** y **puntos de historia**, vamos a describir la **pila del producto** (*Product Backlog*).

La **pila del producto** es una lista ordenada y priorizada (según el valor para el cliente) de todas las historias de usuario que forman parte del proyecto. Es decir, para elaborar la pila del producto tendremos que realizar las siguientes acciones:

1. Analizar los requisitos del producto para el cliente.
2. Describir y detallar las historias de usuario. En este caso usaremos una herramienta (Jira) para definir y gestionar las historias de usuario dentro del método ágil *Scrum*.
3. Estimar las historias de usuario.
4. Asignar una prioridad a cada historia de usuario. Esta acción la realiza el cliente asesorado por el equipo de desarrollo.

## 3.2. Planificación

Una de las mayores diferencias en la planificación con un método ágil respecto a hacerlo con un método tradicional es el concepto de la actividad a realizar. En la planificación de un proyecto tradicional se trata el desarrollo de software como una **actividad totalmente predecible e inmutable**. En la planificación de un proyecto ágil se entiende la actividad a realizar como **actividad flexible y mutable con el paso de las iteraciones**. Se sabe que un proyecto Software de mediano o gran tamaño es una actividad de creación que no se puede evitar que mute considerablemente durante su desarrollo.

Las metodologías tradicionales intentan realizar una estimación objetiva del tamaño total del proyecto a priori. A continuación trazan un plan para ejecutarlo completo y entregar al final del proceso un producto acabado. Las metodologías ágiles dividen el proyecto en pequeñas iteraciones. En cada una de ellas se desarrolla una lista de historias de usuario obteniendo un *Incremento* del producto a entregar. Este producto se le entrega al cliente en cada iteración para recibir su *feedback*. Al terminar cada iteración se repite el proceso hasta agotar todas las historias de usuario. Esto significa que el producto está completamente desarrollado.

Para hacer una estimación aproximada de la duración total del proyecto se suman los puntos de historia de las historias de usuario de la pila inicial del producto. Con este valor y el número

de programadores que van a intervenir en el proyecto se puede hacer una aproximación de la fecha de entrega final del producto.

### 3.3. Estimación de recursos y costes del proyecto

Como se ha comentado en la sección anterior para hacer una estimación total del proyecto se deben sumar los puntos de historia. En nuestro proyecto, como se detalla en el apartado 4.1.1, suman 232 puntos de historia. Prevemos que nuestra velocidad de desarrollo será de 5 puntos de historia diarios (5 horas de trabajo diarias). Obtenemos que la duración total neta de la construcción del producto sera de 232 horas. Por nuestra experiencia añadimos un 7 % de tiempo (16 horas) de margen para reuniones periódicas, descansos, interrupciones e imprevistos. Esto totaliza **248 horas** ( $232 + 16$ ).

A lo largo del proyecto se van a distribuir las historias de usuario en iteraciones o *Sprints*. Como la periodicidad de los *Sprints* será cada 2 semanas y disponemos de 5 horas laborables diarias, cada Sprint tendrá 50 horas ( $5 \text{ horas/día} * 5 \text{ días/semana} * 2 \text{ semanas}$ ). Las 248 horas de desarrollo divididas en *Sprints* de 50 horas da un resultado de 4.96. Redondeado resultan **5 Sprints**.

En la figura 3.2 se muestra la planificación temporal inicial del TFG y la dependencia entre tareas. Se han asignado de forma inicial las historias de usuario del primer *Sprint*. El tiempo del resto de *Sprints* está fijado pero las historias de usuario no se pueden asignar por los posibles cambios del *Product Backlog*. Las horas calculadas para el desarrollo del producto (248 horas) se pueden ver reflejadas en la línea 20 (Desarrollo del producto) de dicha figura.

En el capítulo 6 se puede ver como ha evolucionado a lo largo de las iteraciones el *Product Backlog*.

En la figura 3.3 se puede ver el coste económico el proyecto.



		Nombre	Trabajo	Predec...
1		<b>☐ Proyecto</b>	<b>450 horas</b>	
2		<b>☐ Desarrollo de la propuesta técnica</b>	<b>21 horas</b>	
3		<b>☐ Inicio</b>	<b>5 horas</b>	
4		Definir proyecto con tutor y supervisor	1 hora	
5		Definir método de trabajo y documentac	2 horas	4
6		Definir formato y estándares de trabajo	2 horas	4
7		<b>☐ Documentar y planificar el proyecto</b>	<b>8 horas</b>	
8		Revisar contexto y buscar información	4 horas	3
9		Identificar alcance y objetivos	4 horas	3
10		<b>☐ Planificar el proyecto</b>	<b>8 horas</b>	
11		Definir tareas y estimar fechas	3 horas	7
12		Crear diagrama de Gannt	2 horas	11
13		Documentar la propuesta del proyecto	3 horas	11
14		Entregar propuesta técnica	0 horas	13
15		<b>☐ Desarrollo técnico del proyecto</b>	<b>294 horas</b>	<b>14</b>
16		<b>☐ Definir requisitos del proyecto</b>	<b>14 horas</b>	
17		Definir y documentar requisitos de datos	8 horas	
18		Definir requisitos tecnologicos y de plata	6 horas	
19		<b>☐ Programación</b>	<b>268 horas</b>	
20		<b>☐ Desarrollo del producto</b>	<b>248 horas</b>	
21		<b>☐ Primer sprint</b>	<b>48 horas</b>	
22		Contratos obligacion	16 horas	18
23		Mejoras de Contrato	16 horas	22
24		Expedientes de Contrato	16 horas	23
25		Segundo sprint	50 horas	
26		Tercer sprint	50 horas	
27		Cuarto sprint	50 horas	
28		Quinto sprint	50 horas	
29		Pruebas de integración	20 horas	28
30		<b>☐ Puesta en marcha</b>	<b>12 horas</b>	<b>29</b>
31		Implantación	8 horas	29
32		Formacion	4 horas	31
33		Entrega final	0 horas	32
34		<b>☐ Documentación y presentación del TFC</b>	<b>135 horas</b>	<b>33</b>
35		Redacción de informes quincenales	4 horas	14
36		Redacción de la memoria técnica	100 horas	29
37		Entrega e la memoria técnica	0 horas	36
38		Preparación de la presentación oral	30 horas	37
39		Presentación oral	1 hora	38

Figura 3.2: Planificación temporal inicial del proyecto y las dependencias entre tareas junto con la asignación inicial de historias de usuario del primer Sprint

tareas	horas/persona
Contextualizar el proyecto	8
Planificar el proyecto	8
Desarrollo técnico del proyecto	294
	310

Constantes de cálculo	
Horas de trabajo diarias	5
Días de trabajo mensual	20
Coste mensual por persona (horario completo)	2500
Coste mensual por persona (5 horas diarias)	1562,5

Personas asignadas al proyecto	1
Meses invertidos en el proyecto	3,1
% incremento del proyecto por costes indirectos	20%
Coste total del proyecto	5.812,50 €

Figura 3.3: Coste económico del proyecto.

En la figura 3.4 se muestra el diagrama de Gantt que ilustra la temporización del proyecto

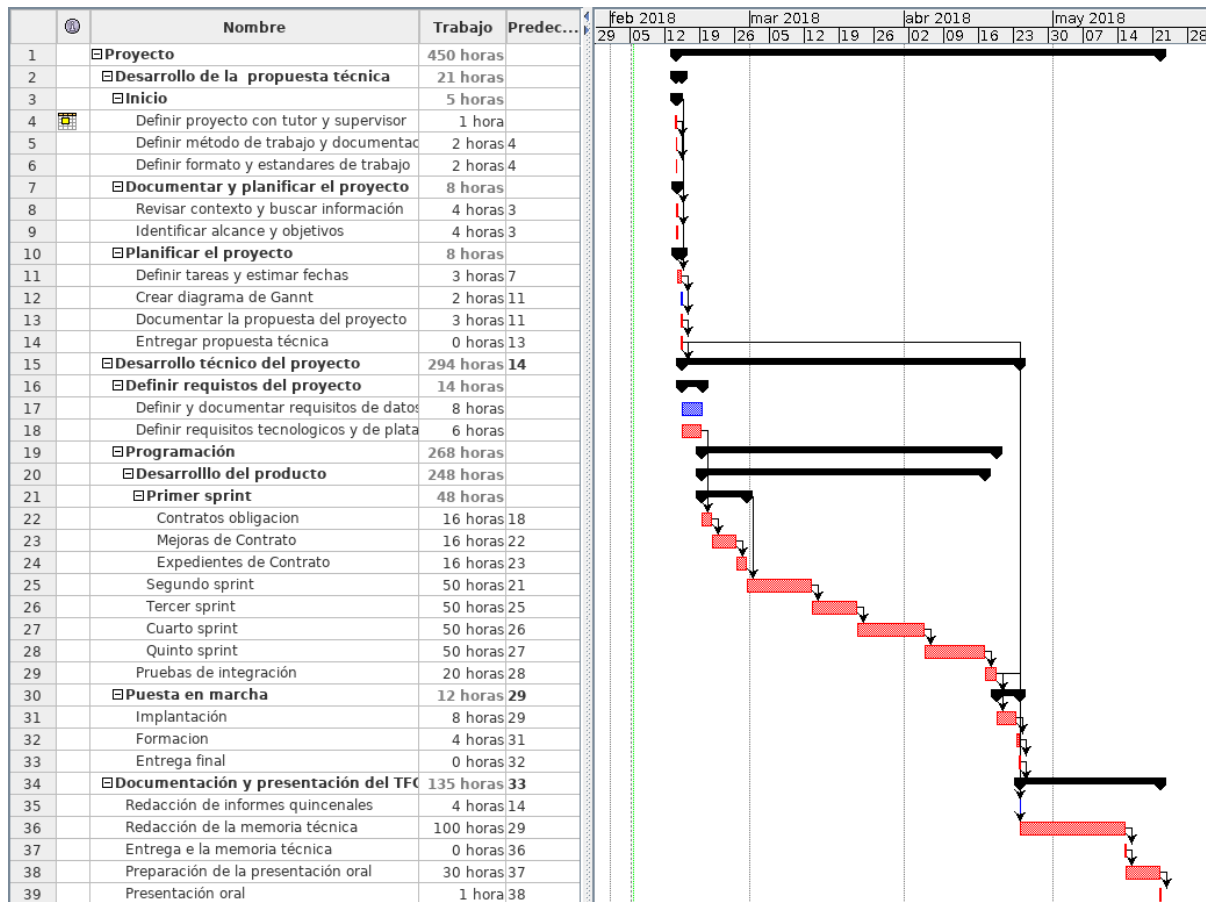


Figura 3.4: Diagrama de Gantt con la temporización del proyecto.

### 3.4. Seguimiento del proyecto

El seguimiento del proyecto se realizará de las siguientes formas:

- El Supervisor del proyecto dirigirá y revisará el proyecto. El uso de *Git flow* como protocolo de supervisión hace que toda modificación de código pase por su revisión.
- La metodología elegida (*Scrum*) obliga a un control diario (*Daily Scrum*) por el resto de los desarrolladores.
- La metodología elegida (*Scrum*) obliga a un control cada dos semanas (revisión del *Sprint* de duración dos semanas) por parte de todos los involucrados en el proyecto.

## Capítulo 4

# Análisis y diseño del sistema

En este apartado se explica tanto el análisis *Scrum* del proyecto como el diseño de todos sus aspectos.

### 4.1. Análisis del sistema

En el apartado 3.1.3 hemos mencionado el análisis *Scrum* hace innecesario iniciar el análisis con un diagrama de casos de usos. A pesar de ello y a título meramente ilustrativo incluimos en la figura 4.1 un diagrama de casos de usos del proyecto.

#### 4.1.1. Pila de producto inicial (*initial product backlog*)

La lista inicial del *Product Backlog* esta formada por las historias de usuario enumeradas a continuación. El punto de historia es aproximadamente equivalente a 1 hora de trabajo. La prioridad asignada por el cliente puede ser Máxima, Alta, Media, Baja o Muy baja.

- **HU01** Como usuario necesito visualizar la información de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Máxima].
- **HU02** Como usuario necesito visualizar la información las mejoras de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Alta].
- **HU03** Como usuario necesito visualizar la información de los expedientes de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Máxima].
- **HU04** Como usuario necesito visualizar los pesajes de residuos por tipo de residuo relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].

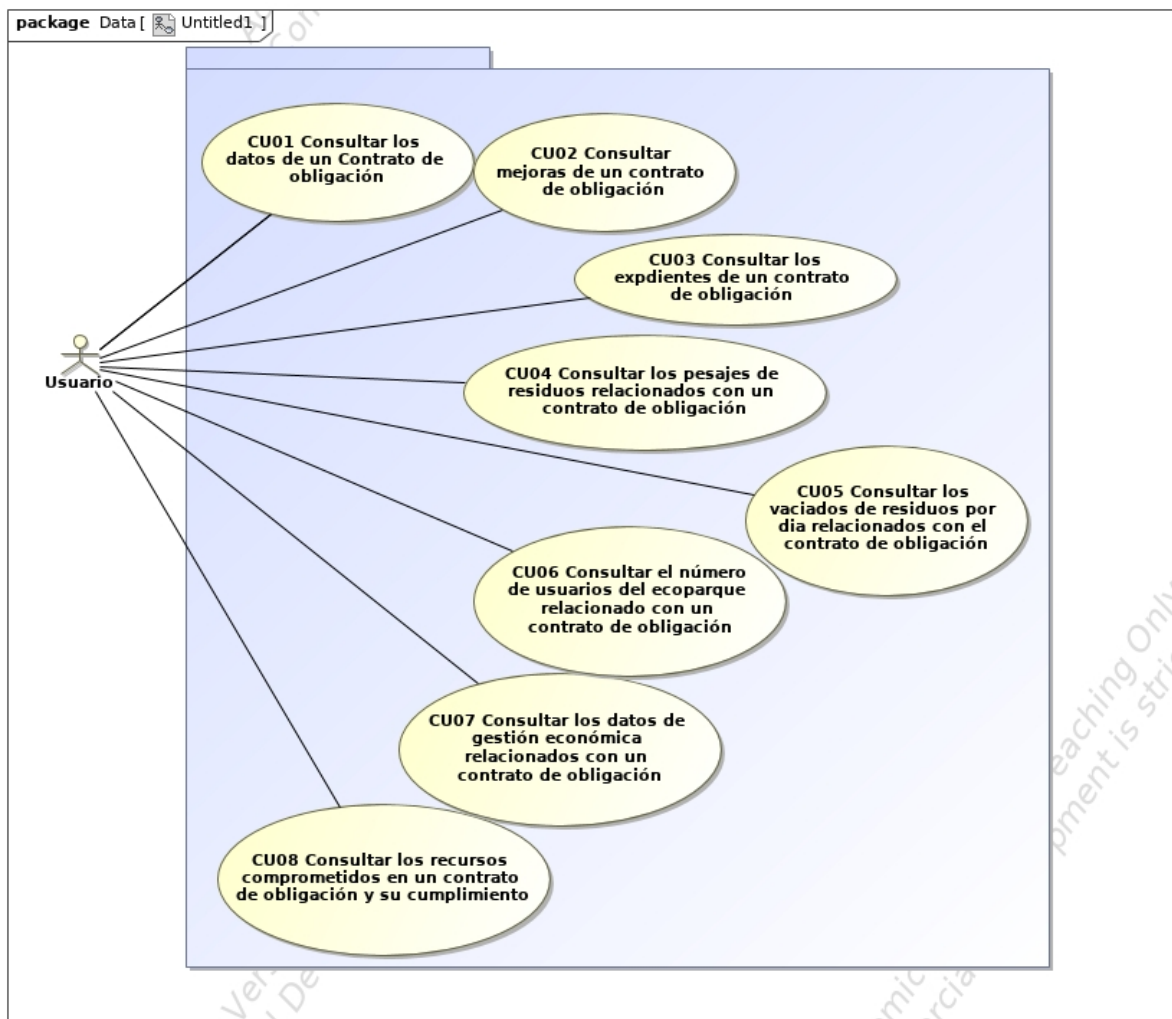


Figura 4.1: Diagrama de casos de uso del proyecto (a título ilustrativo).

- **HU05** Como usuario necesito visualizar los vaciados de contenedores de residuos por fecha relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU06** Como usuario necesito visualizar los datos de gestión económica relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU07** Como usuario necesito visualizar la información de las horas comprometidas en los expedientes de contratos de obligación para poder consultar su información [48 puntos de historia] [Prioridad Media].
- **HU08** Como usuario necesito visualizar la información de las horas comprometidas y cumplidas en los expedientes de contratos de obligación para poder consultar su información [48 puntos de historia] [Prioridad Media].
- **HU09** Como usuario necesito visualizar el número de usuarios que han hecho uso del eco-parque relacionado con el contrato de obligación para poder consultar su información



[16 puntos de historia] [Prioridad Baja].

En la figura 4.2 se puede ver el *backlog* completo inicial usando Jira.

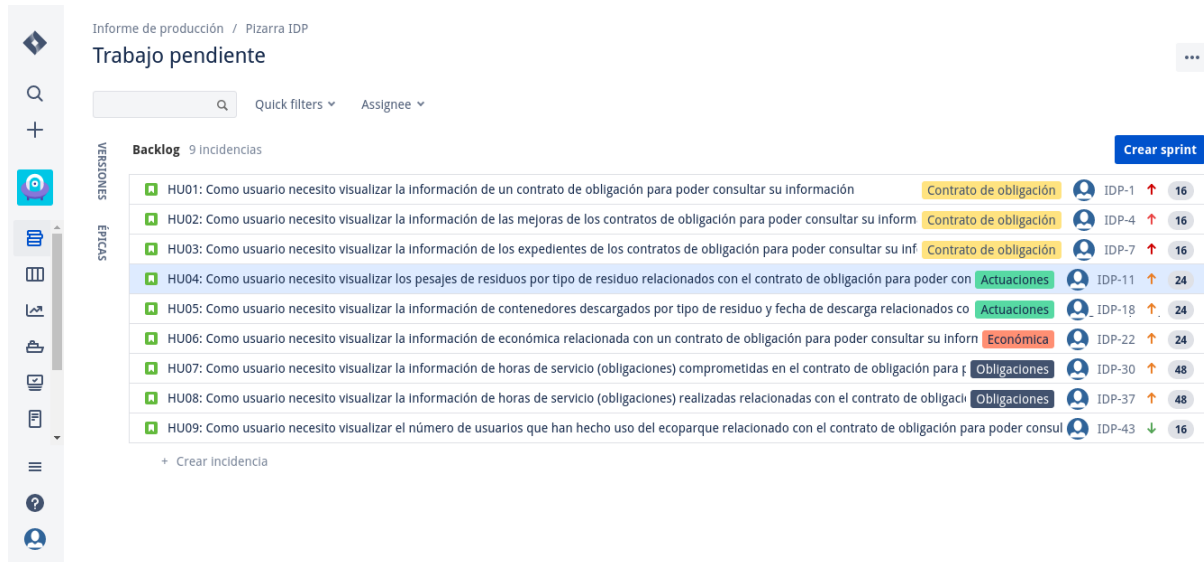



Figura 4.2: Pila de producto (*Product Backlog*) inicial.

#### 4.1.2. Pila de producto final

Aunque no corresponde al Análisis del sistema incluimos como ha quedado la pila de producto después de las iteraciones. La intención es que quede mejor ilustrado el proceso para el lector. Las historias de usuario en

- verde han quedado completadas.
  - rojo están pendientes de implementar.
  - azul están parcialmente completadas.
  - magenta las añadidas o modificadas.
  - magenta las eliminadas.
- 
- **HU01** Como usuario necesito visualizar la información de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Máxima].
  - **HU02** Como usuario necesito visualizar la información las mejoras de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Alta].
  - **HU03** Como usuario necesito visualizar la información de los expedientes de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Máxima].

- **HU04** Como usuario necesito visualizar los pesajes de residuos por tipo de residuo relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU05** Como usuario necesito visualizar los vaciados de contenedores de residuos por fecha relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU06** Como usuario necesito visualizar los datos de gestión económica relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU07** Como usuario necesito visualizar la información de las horas comprometidas en los expedientes de contratos de obligación para poder consultar su información [48 puntos de historia] [Prioridad Media].
- **HU08** Como usuario necesito visualizar la información de las horas cumplidas en los expedientes de contratos de obligación para poder consultar su información [48 puntos de historia] [Prioridad Media].
- ~~**HU09** Como usuario necesito visualizar el  de usuarios que han hecho uso del eco-parque relacionado con el contrato de obliga para poder consultar su informa [16 puntos de historia] [Prioridad Baja].~~

La historia de usuario HU09 ha sido finalmente descartada por el cliente. La razón es no disponer de un origen de datos para alimentar esta información.

## 4.2. Diseño del sistema

En esta sección se especifica el diseño datos, de la arquitectura del sistema y de la interfaz gráfica de usuario.

### 4.2.1. Diseño de datos

En la figura 4.3 se muestra el diagrama de dependencias de las clases del modelo del proyecto.

En la figura 4.4 se muestra el modelado con atributos de clases del modelo del proyecto.

### 4.2.2. Diseño de la arquitectura

La arquitectura del sistema informático representa la estructura de los componentes del sistema y la interacción entre ellos. La decisión sobre el tipo de arquitectura elegido se basa en el cumplimiento de todos los requerimientos del proyecto. Por tanto este proceso es crucial para el éxito del proyecto.

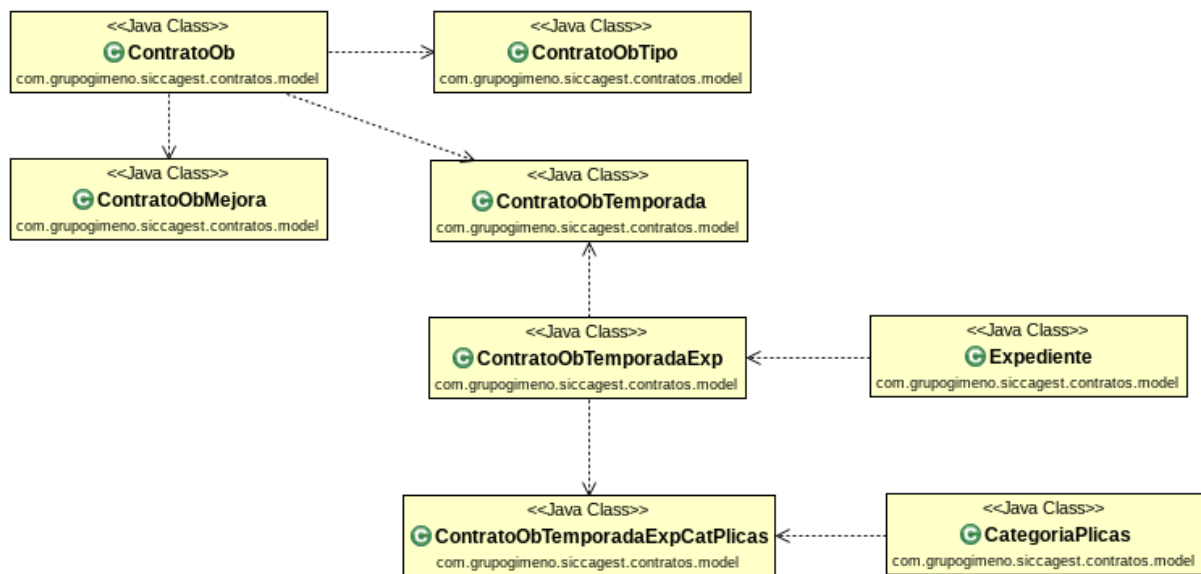


Figura 4.3: Diagrama de dependencias de las clases del modelo del proyecto.

la arquitectura centrada en los datos se constituye alrededor de un almacén de datos y sus diferentes representaciones en cada una de las capas de la aplicación.

En la arquitectura orientada a objetos los componentes encapsulan los datos y las operaciones para manipularlos. La comunicación entre componentes es por medio de paso de mensajes.

Finalmente el sistema presentará una arquitectura cliente/servidor sin estado. Los clientes acceden por HTTP (protocolo sin estado) a un servidor para obtener información. Existe, por tanto, un servidor que contiene el almacén de datos y que sirve a una gran cantidad de clientes que se conectan a él para la obtención de datos.

#### 4.2.2.1. Estilo arquitectónico

Los patrones o estilos arquitectónicos básicos en el diseño del sistema son la **arquitectura centrada en los datos**, **arquitectura orientada a objetos** y la **arquitectura cliente/servidor**

#### 4.2.2.2. Arquitectura del sistema

En la figura 4.5 se puede ver la arquitectura del sistema durante el desarrollo.

En la figura 4.6 se representa la arquitectura del sistema a desarrollar en ejecución.

#### 4.2.2.3. Arquitectura de la aplicación web

En la aplicación se ha decidido utilizar el patrón MVC (Modelo-Vista-Controlador). El principal objetivo es separar e independizar al máximo con el fin de poder mantener de forma independiente:

- **El modelo:** Está en los módulos de representación de los datos (paquete model del proyecto de Backend) y de acceso a la persistencia de los datos (paquete daos). La persistencia de datos se encuentra en un almacén de datos SQLServer para Contratos de obligación y Partes de trabajo y en otro almacén de datos transparente para la aplicación en lo referente a pesajes y contenedores.
- **Vista:** Se encuentra localizada en los módulos de interfaz de usuario (paquete frontend del proyecto de Frontend), generados con Angular y que contienen tanto la representación gráfica en HTML como la lógica de visualización en TypeScript.
- **Controlador:** contiene la lógica de negocio de la aplicación (paquete java del proyecto de FrontEnd). Es el módulo encargado de mediar entre la Vista y el Modelo. Recibe las peticiones de datos de la Vista, solicita al Modelo esta información y se la transmite a la Vista, tras el procesamiento adecuado.

Aplicando este patrón conseguimos varios beneficios:

- Disponer de 3 módulos independientes y reemplazables individualmente.
- Reutilización de código.
- Facilitar el desarrollo de la aplicación.
- Permitir un más fácil mantenimiento.
- Crecer en escalabilidad.

#### 4.2.3. Diseño de la interfaz de usuario (GUI)

El interfaz de usuario es el canal por el que el usuario se comunica con la máquina. Comprende los puntos de contacto entre el usuario y el equipo, y debe seguir las siguientes reglas:

- Proporcionar una interacción flexible.
- Permitir una interacción directa con los objetos que se muestran en la pantalla.
- Permitir que el usuario interrumpa o deshaga acciones.
- Comunicar al usuario la finalización correcta de sus peticiones.
- Comunicar de formar inteligible cualquier suceso inesperado en el sistema.

- El formato visual debe adaptarse al usuario y sus limitaciones.
- Organizar de una forma lógica los datos a mostrar.
- Mantener la consistencia de la interfaz.

Es de una extrema importancia realizar un diseño de la interfaz gráfica del sistema antes de la fase de construcción del mismo. Esto por diversos motivos:

- Tener una referencia visual del resultado para todos los desarrolladores.
- Construir conjuntamente con el cliente final la presentación de la aplicación.
- Validar con el cliente que se cumplan sus expectativas.

#### **4.2.3.1. Interfaz gráfica**

En las figuras 4.7, 4.8, 4.9 y 4.10 se pueden ver respectivamente los prototipos de pantalla de detalle del Contrato de obligación, de sus obligaciones, de sus actuaciones y de su gestión económica.

Los prototipos han sido generados con el asesoramiento del experto en diseños de interfaces de usuario de ADC Infraestructuras y Sistemas. La razón es que la aplicación a desarrollar deberá integrarse en el futuro en el portal de aplicaciones de la empresa. No tenemos, por tanto, la libertad de realizar el diseño que al equipo desarrollador le parezca más apropiado.



Figura 4.4: Diagrama de clases con atributos del modelo del proyecto.

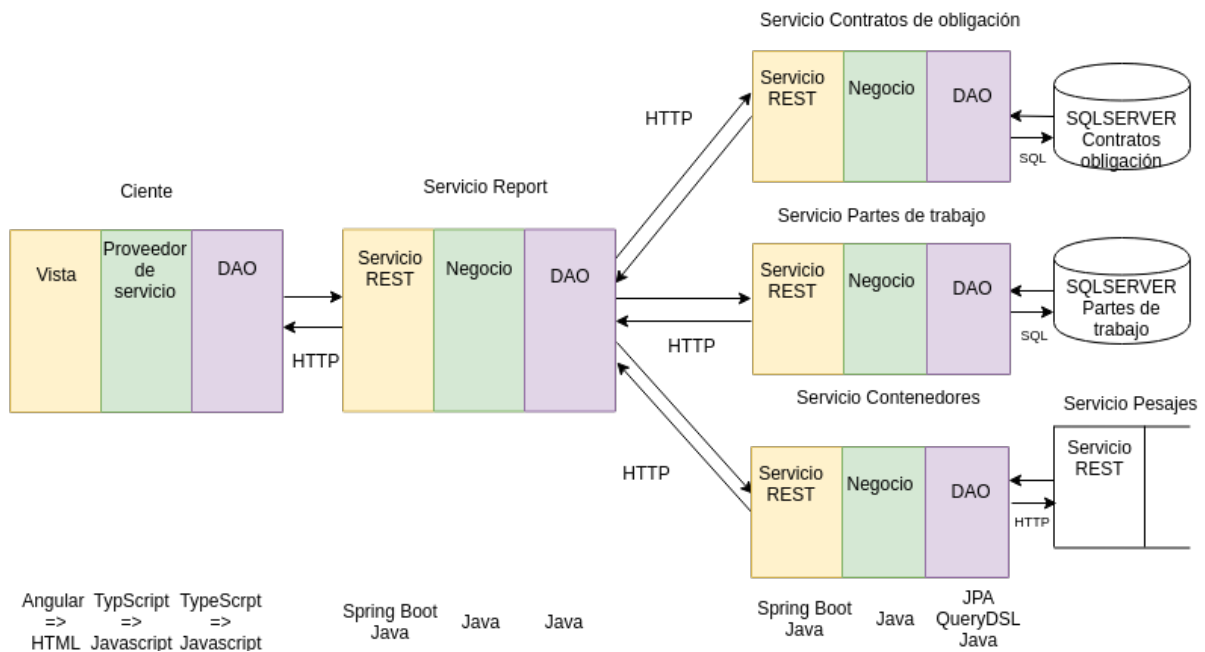


Figura 4.5: Arquitectura software del sistema en desarrollo.

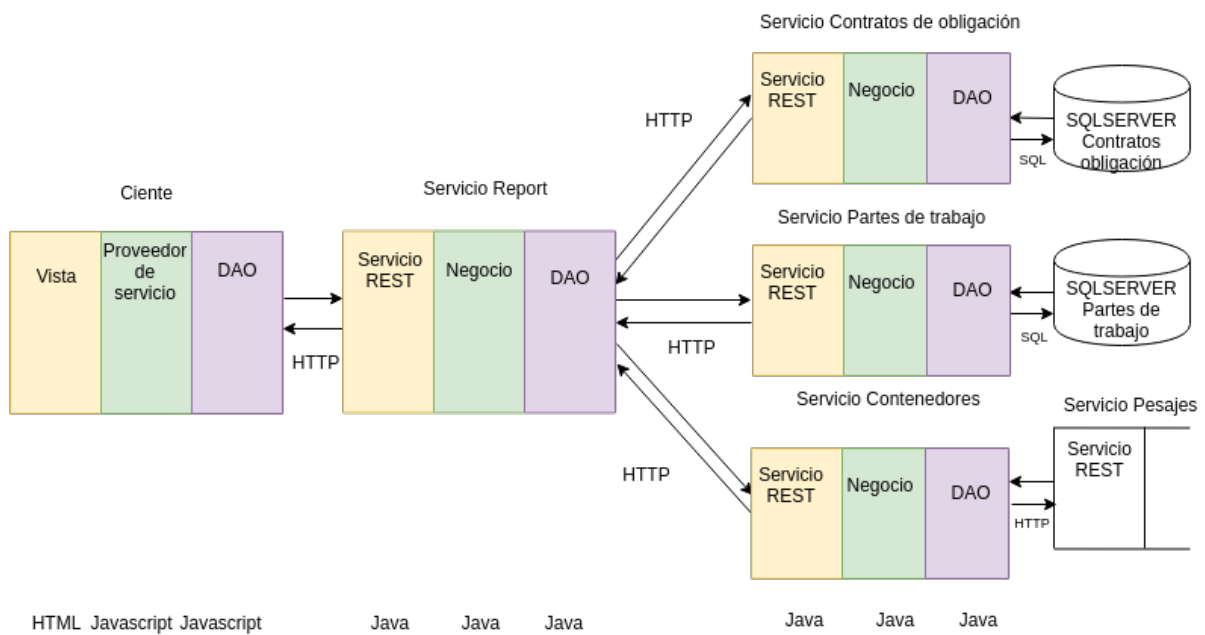


Figura 4.6: Arquitectura software del sistema en ejecución.

Detalles del contrato

Obligaciones

Actuaciones

Gestión económica

---

**Nº contrato 18604804860**

Fecha inicio

Fecha fin

Campo

Campo

Mejoras

Descripción	Total €/año	Total €/contrato	Coste real	Puntuación	Realizada

Figura 4.7: Prototipo de la pantalla de detalle del Contrato de obligación.



Detalles del contrato

Obligaciones

Actuaciones

Gestión económica

**N° contrato 18604804860**

Periodo

Expediente Benicasim RSU

Recursos	H. Programadas	H. Realizadas	Total dif. horas

Expediente Benicasim LV

Recursos	H. Programadas	H. Realizadas	Total dif. horas

Figura 4.8: Prototipo de la pantalla de obligaciones del Contrato de obligación.

Detalles del contrato

Obligaciones

Actuaciones



Gestión económica

N° contrato 18604804860

Periodo

Pesajes

Vaciado de contenedores

Residuo	Fecha	N° contenedores levantados	Peso

Figura 4.9: Prototipo de la pantalla de actuaciones del Contrato de obligación.

Detalles del contrato

Obligaciones

Actuaciones

Gestión económica

N° contrato 18604804860

Periodo

Tipo coste	Descripción	Importe

Figura 4.10: Prototipo de la pantalla de gestión económica del Contrato de obligación.



## Capítulo 5

# Descripción de la implementación y pruebas

En esta sección se describen las tecnologías, herramientas y lenguajes usados para la construcción de la aplicación, los módulos en los que se ha distribuido el código, los patrones de diseño utilizados y los tipos de pruebas realizadas.

### 5.1. Lenguajes y herramientas utilizado en la implementación

Los lenguajes usados para el desarrollo del proyecto son:

- **Java:** Lenguaje de alto nivel y propósito general empleado en todas las capas de los diferentes servicios REST que forman parte de la aplicación.
- **TypeScript:** Lenguaje de alto nivel y propósito general empleado en las capas Proveedor de servicios y DAO del proyecto de Fontend.
- **JavaScript:** Lenguaje al que se compila TypeScript por ser el único que puede ejecutarse de forma estándar en todos los navegadores.
- **HTML *Hypertext Markup Language*:** Lenguaje de marcas que puede ejecutarse de forma estándar en todos los navegadores. Con él (enriquecido con etiquetas de las que nos provee Angular) se construye la interfaz de usuario.
- **CSS:** Hojas de estilos que aportan riqueza a la visualización de elementos HTML.

Las herramientas para el desarrollo del proyecto son:

- **Angular:** Framework que ayuda a desarrollar aplicaciones HTML con JavaScript embebido.

- **PrimeNG**: Librería de componentes contruidos para su uso con Angular.
- **IntelliJ**: IDE de desarrollo.
- **Maven**: Herramienta de software para la gestión y construcción de proyectos Java.
- **Spring**: Framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java. En el proyecto hacemos uso de la variante *Spring boot*.
- **JUnit**: Framework que facilita la construcción de pruebas unitarias y de integración.
- **Git**: Software de control de versiones.
- **Gitlab**: Servicio web de control de versiones y desarrollo de software colaborativo basado en Git.
- **Jira**: Software de gestión y seguimiento de trabajo en equipo.
- **Jenkins**: Software de integración continua escrito en Java.
- **JPA *Java Persistence Api***: API desarrollada para JavaEE que facilita el acceso a bases de datos relacionales.
- **QueryDSL**: Framework que facilita la construcción y mantenimiento de consultas a bases de datos relacionales.
- **TSLint**: Herramienta de análisis de código TypeScript, que asegura la calidad del código generado.

## 5.2. Proyectos y módulos de software

El desarrollo del proyecto completo se ha distribuido en 2 proyectos Maven:

- Proyecto **siccagest-contratos-obligacion-service**: Contiene la generación y despliegue de los servicios REST de Contratos de obligación (centrado en la entidad Contrato de obligación), de Partes de trabajo (centrado en la entidad Parte de trabajo) y de Información Económica (centrado en la entidad Información económica). Todos estos servicios son genéricos y utilizable por cualquier otro proyecto que necesite el acceso a las mismas entidades. No tienen nada específico respecto a la presentación o tratamiento de datos para la presentación al cliente.
- Proyecto **fobesa-fobasa-web-report**: Contiene 2 grandes módulos:
  - Servicio REST de Report (centrado en la entidad Report). Este servicio accede a los otros 3 servicios REST mencionados en el punto anterior y a otro servicio REST de consulta de Pesajes y Contenedores. Contiene la lógica de negocio específica de la aplicación y hace un tratamiento de datos para facilitar la presentación de la información para la Vista.

- Modulo de generación de la Vista. Utiliza Angular (con HTML y TypeScript) para generar HTML y Javascript. Además dispone de una capa de servicio para llamar al servicio REST de Report.

Cada uno de los servicios REST consta de la siguiente estructura en módulos:

- **Service.** Es una capa que se encuentra en el paquete *service*. En el está el servicio REST y el controlador REST. Ambos son desplegados por *Spring* en el servidor.
- **DAOs.** Es una capa que se encuentra en el paquete *daos*. Es la capa de acceso a la persistencia.
- **Model.** No es una capa. Es un modulo con clases POJO (*Plain Old Java Object*) del modelo de datos, que comparten todas las capas del servicio.

El modulo que genera la vista dispone de la siguiente estructura:

- **Components :** Contiene los componentes visuales creados con Angular.
- **DAOs :** Contiene la capa de acceso a los servicios de backEnd.
- **Services .** Contiene la capa de servicios que accede a la capa daos.
- **Model.** No es una capa. Es un modulo con clases POJO (*Plain Old Java Object*) del modelo de datos,

Especial mención merece a nivel de módulos de software la estructura de tests. Todos los proyectos tienen un paquete test que contiene las pruebas unitarias, de integración y aceptación. En el caso de pruebas de aceptación están diseñadas de forma que al subir un cambio a la rama develop o master del repositorio git se desencadena la ejecución de los test. La aplicación de integración continua Jenkins se encarga de que si los test no son satisfactorios no se produce la publicación de la aplicación.

En la figura 5.1 se puede ver la lista de intentos satisfactorios e insatisfactorios de despliegue de la aplicación.

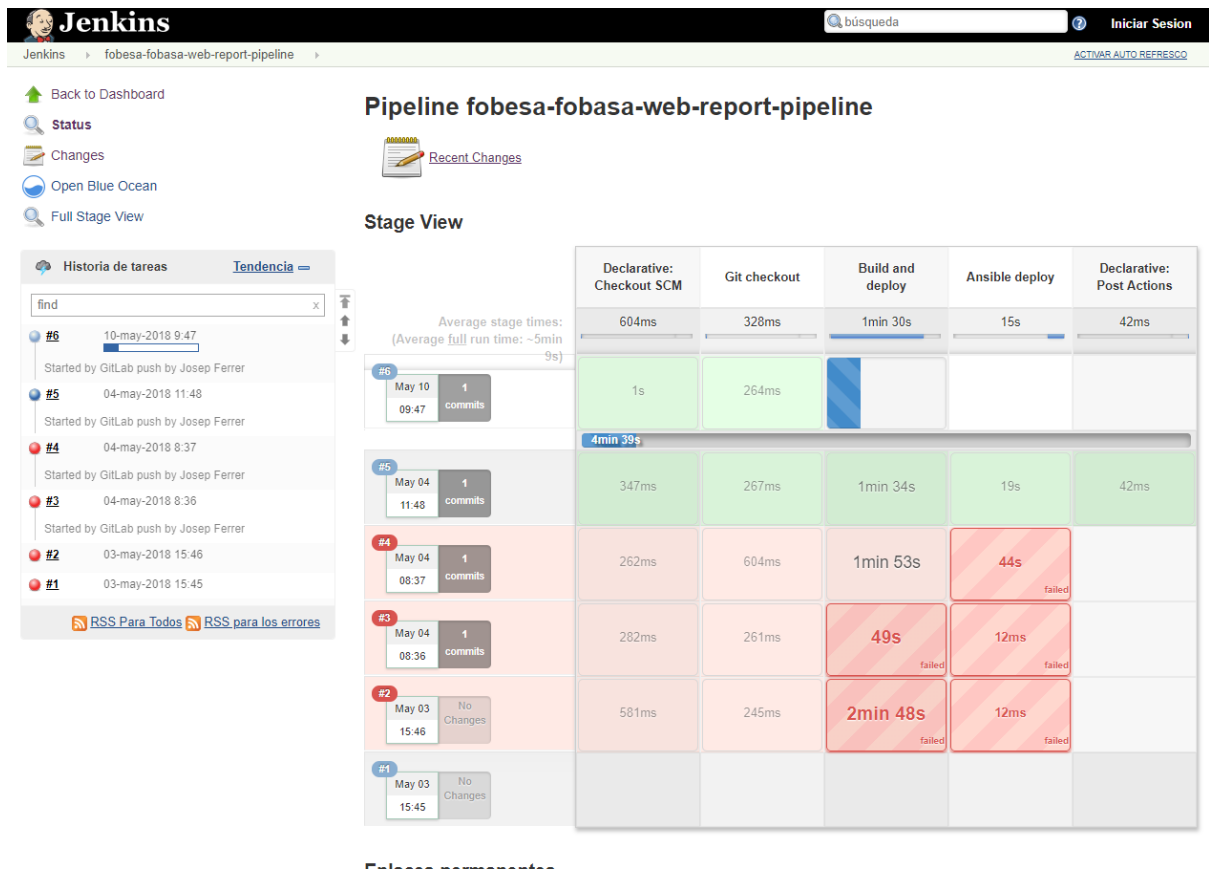


Figura 5.1: Ejemplo de pantalla de despliegue de la aplicación en Jenkins.

## 5.3. Patrones de diseño

Los patrones de diseño son soluciones genéricas y probadas para problemas típicos y recurrentes que podemos encontrarnos al desarrollar una aplicación. A continuación describimos los principales patrones utilizados en la implementación de la aplicación desarrollada.

### 5.3.1. Patrón Singleton y su variante Multiton

Se trata de un patrón de diseño de la categoría creación. El principal objetivo es asegurarse de que solo existe una instancia de una determinada clase y proporcionar un punto de acceso único a dicha instancia. En nuestra aplicación se usa una variante de este patrón llamada Multiton. En esta variante en lugar de tener una solo instancia se dispone de un pool limitado de instancias, que son reutilizadas continuamente.

En la figura 5.2 (extraída de la página (Patrón Singleton) [9]) se muestra la estructura de dicho patrón, así mismo en la figura 5.3 (extraída de la página (Patrón Multiton) [7]) se puede ver la estructura de dicha variante.





Figura 5.2: Patrón Singleton.

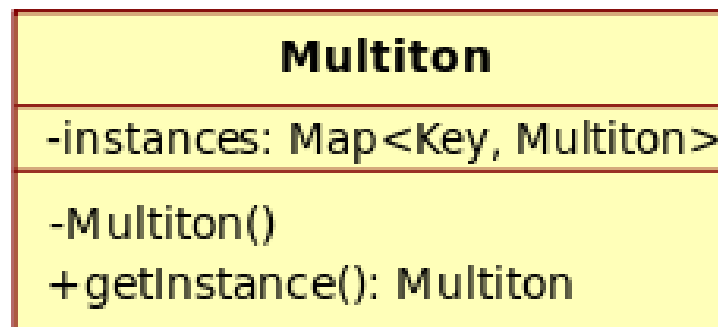


Figura 5.3: Patrón Multiton.

En el proyecto este patrón es usado en:

- Todas las clases de acceso a base de datos tienen un atributo EntityManager. Esta clase se encarga de crear la conexión JDBC (*Java Database Connectivity*) a la base de datos y reutilizarla siempre que sea posible.

### 5.3.2. Patrón Builder

Se trata de otro patrón de creación. Su objetivo es la construcción de objetos sin tener en cuenta su representación, de modo que el mismo proceso de construcción permite crear diferentes representaciones. El mayor interés de este patrón es eliminar dependencias fuertes entre clases. La clase que hace uso del Builder no tiene dependencia, al menos fuerte, de la clase construida.

En la figura 5.4 (extraída de la página (Patrón Builder) [5]) se muestra la estructura de dicho patrón.

En el proyecto este patrón es usado en:

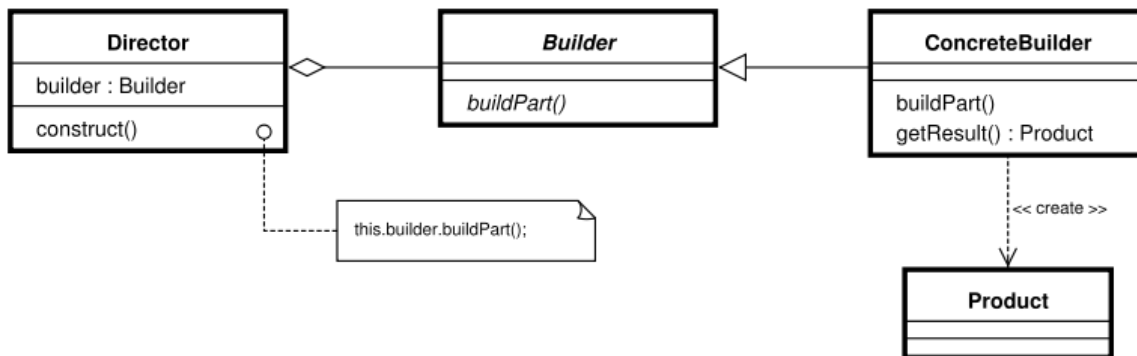


Figura 5.4: Patrón Builder.

- TimestampBuider es una clase que construye objetos Timestamp descargando a la clase que lo usa de la complejidad del manejo de formatos de fecha.
- ReportBuider es una clase que construye objetos Report para ejecutar los test de integración.
- ContratoBuider es una clase que construye objetos Contrato para ejecutar los test de integración.
- ObligacionesBuider es una clase que construye objetos Obligación para ejecutar los test de integración.
- EconomicaBuider es una clase que construye objetos InformacionEconomica para ejecutar los test de integración.

### 5.3.3. Patrón Dependency injection

Es un patrón de diseño orientado a objetos, en el que se suministran objetos instanciados a una clase que los precisa en lugar de ser la propia clase la que cree dichos objetos. Esos objetos cumplen contratos que conoce la clase que los usa para poder funcionar (de ahí el concepto de dependencia). Nuestras clases no crean los objetos que necesitan, sino que se los suministra otra clase 'contenedora' que inyectará la implementación deseada.

En la figura 5.5 (extraída de la página (Patrón Inversión de dependencias) [6]) se muestra la estructura de dicho patrón.

En el proyecto este patrón es usado en:

- Todas las instancias de objeto marcadas con la etiqueta *@Autowired* de *Spring*. *Spring* se encarga de crear una instancia que cumpla la interface definida por nosotros e inyectarla en la clase que la requiera.

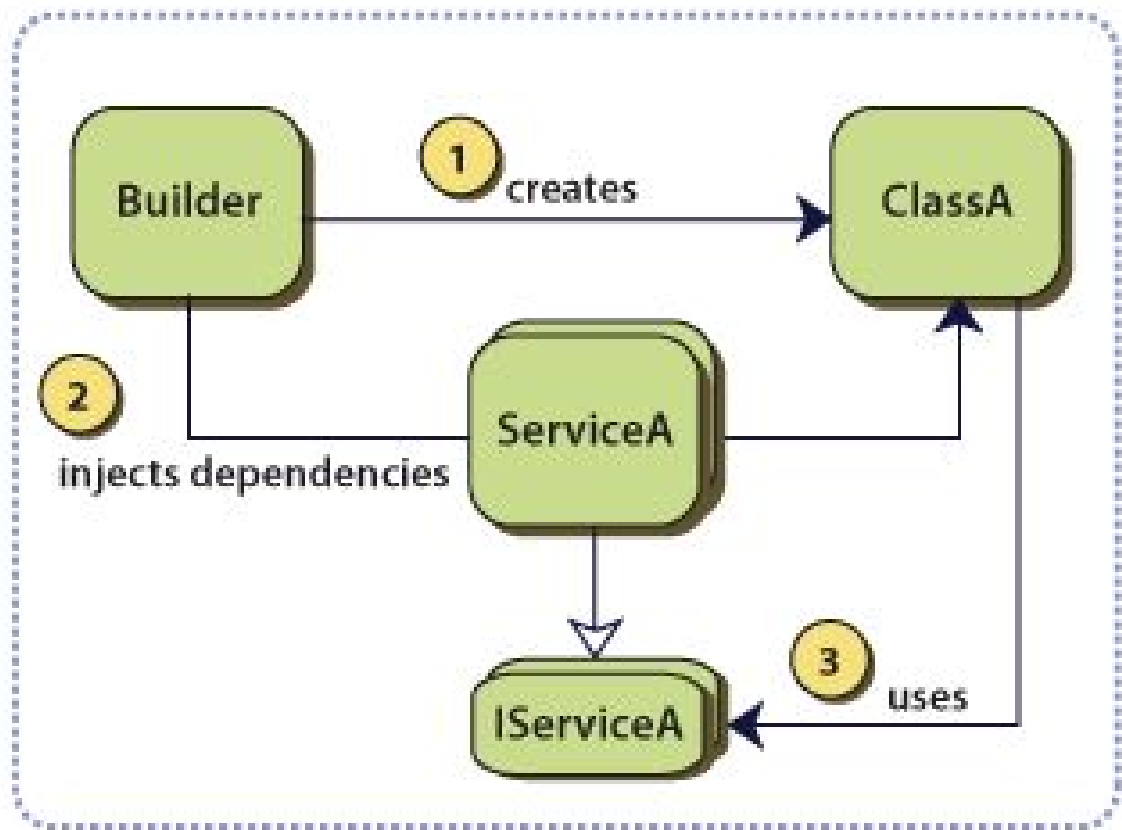


Figura 5.5: Patrón Inversión de dependencias.

#### 5.3.4. Patrón Observer

Define una dependencia del tipo uno a muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, notifica este cambio a todos los dependientes. Se trata de un patrón de comportamiento.

En la figura 5.6 (extraída de la página (Patrón Observer) [8]) se muestra la estructura de dicho patrón.

En el proyecto este patrón es usado en:

- La relación entre los componentes visuales HTML y los atributos de objetos Javascript. Los componentes visuales HTML se suscriben a las modificaciones de contenido de un atributo de una clase Javascript. Con ello al modificar el contenido de este atributo, el nuevo contenido se refleja en el componente HTML.

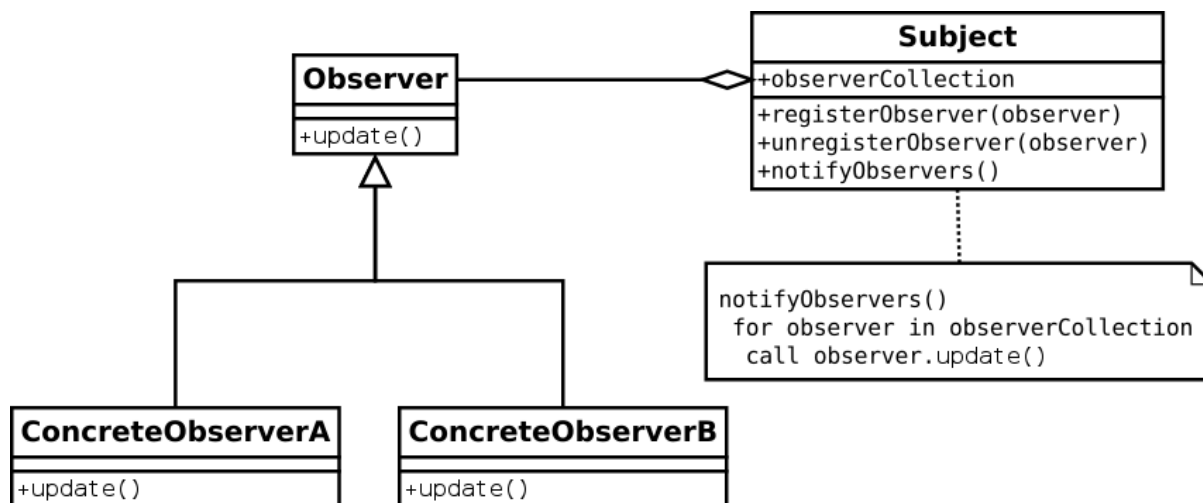


Figura 5.6: Patrón Observer.

## 5.4. Pruebas unitarias

Las pruebas unitarias o tests unitarios tienen como objetivo comprobar el buen funcionamiento de un módulo de software independiente. Este tipo de pruebas debe de seguir los principios **FIRST** (*Fast, Independent, Repeteable, self-validating* y *timely*). Estos principios recomiendan escribir pruebas rápidas de ejecutar, independientes unas de otras, repetibles con las mismas condiciones de entrada, que se valide el resultado en la propia prueba y escritas antes o durante el desarrollo del código.

Para el diseño de las pruebas unitarias se ha usado **JUnit**. Se trata de un *framework* de código abierto que permite escribir pruebas unitarias para múltiples lenguajes, incluido Java.

En el proyecto se han desarrollado pruebas unitarias:

- ExpedientesTest. Ubicado en el servicio REST más cercano a la vista. Calcula los expedientes a contemplar, dado un número de Contrato y las fechas desde y hasta.
- InformacionEconomicaTest. Comprueba, en todos los casos, que la acumulación de datos económicos totalizados por clase de coste es correcta.
- ObligacionesTest. Comprueba, en todos los casos, que la acumulación de obligaciones programadas y reales totalizadas por categoría de plicas (tipo de recurso) es correcta.

## 5.5. Pruebas de aceptación

Las pruebas de aceptación se encargan de verificar que un producto cumpla con los estándares necesarios y satisfaga los requisitos establecidos por los usuarios.

El desarrollo del producto se ha distribuido en varias iteraciones o *Sprints*. En cada reunión

de planificación del *Sprint* se han diseñado las pruebas de aceptación de cada historia de usuario. Al finalizar el *Sprint* se pasan las pruebas de aceptación. Si el sistema satisface las pruebas la historia de usuario puede darse por implementada. En caso contrario en la siguiente iteración debe resolverse la incidencia.

En este punto es de especial importancia comentar el uso de **Jenkins** en la estructura de desarrollo de software. Jenkins es un software open-source escrito en Java. Jenkins proporciona integración continua para el desarrollo de software. Cada vez que se modifica la rama Master o Development, Jenkins se encarga de publicar en Producción o Desarrollo una nueva versión de la aplicación. Antes de hacerlo comprueba que el software completo satisfaga todas las pruebas de aceptación. En caso de no satisfacerlas no se produce la publicación. Con esto queda garantizado que ninguna modificación violará ninguna de las pruebas de aceptación diseñadas.

Las pruebas de aceptación tienen la siguiente estructura:

**DADO** un escenario (condiciones previas)

**CUANDO** se produzca un determinado evento

**ENTONCES** se produce una salida o una modificación del estado del sistema

Para el diseño de las pruebas de aceptación se han seguido los principios **INVEST** para la definición de *users stories*:

- **Independent**: se desarrollan y prueban de forma independiente unas de las otras.
- **Negotiable**: resultado del análisis conjunto de todas las partes implicadas en el proyecto.
- **Valuable**: tiene valor de negocio para el cliente.
- **Estimable**: debe poderse estimar el esfuerzo que requiere su desarrollo.
- **Small**: debe poder completarse en una sola iteración o *Sprint*.
- **Testeable**: el cliente/usuario/desarrollador debe poder confirmar que se ha completado satisfactoriamente.

Las pruebas aceptación, integración y unitarias usan una base de datos H2 ejecutándose localmente. En el anexo C se pueden ver las sentencias que rellenan la base de datos para ejecutar los test.

## 5.6. Aseguramiento de la calidad del código

Con el fin de garantizar la calidad del código generado nos hemos apoyado en varias herramientas:

- *TSLint*: es una herramienta de análisis estático del código que comprueba el código TypeScript sobre cumplimiento de reglas, mantenibilidad y evitar errores funcionales. Toda

la información sobre la herramienta está en (TSLint) [10].

En el archivo del proyecto tslint.json se configura aquello que queremos que compruebe la herramienta. Para no que no sea demasiado extenso, a modo de ejemplo: para comprobar que cada línea se acaba en punto y coma:

```
"semicolon": ["always"]
```

- *GG\_SENDA*: Configuración de la organización para garantizar la calidad del código Java y homogeneizarlo. Su configuración se encuentra en el fichero GG\_SENDA.xml. A nivel de ejemplo: para comprobar que no queden imports sin uso, se incluye:  
«inspection\_tool enabled\_by\_default="true" level="ERROR" enabled="true" class="UnusedImport"»  
Un ejemplo de este análisis se puede ver en la figura 5.7.

- *Calculate metrics*: opción de IntelliJ que calcula diversos tipos de métricas, como se puede ver en la figura 5.8

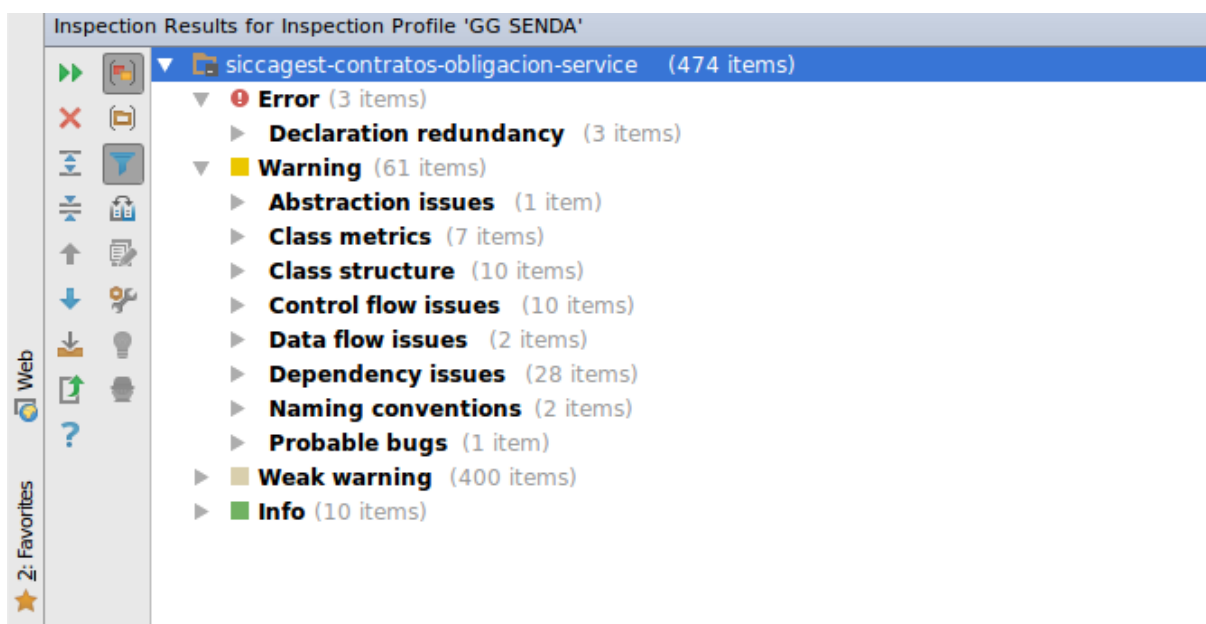


Figura 5.7: Inspección de código propio de la organización.

Figura 5.8: Metricas de analisis estatico de IntelliJ.





## Capítulo 6

# Construcción e Iteraciones

### 6.1. *Sprint* 1

El primer *Sprint* se lleva a cabo en la quincena del 05-03-2018 al 16-03-2018.

#### 6.1.1. Planificación

En la reunión de planificación del *Spring* 1 participan la figura del *Scrum Master* (representada por el Supervisor del TFG), el *Product Owner* (representada por la Responsable de Software de ADC) y el equipo desarrollador (representado por Josep Ferrer).

En ella se toman decisiones críticas para el desarrollo del *Sprint*. Las más importantes, en este caso, son:

- El *Product Owner* acepta el prototipo de la figura 4.7 como el resultado visual que desea.
- Todos los asistentes consensúan las Historias de usuario a incluir en el *Sprint*: HU01, HU02 y HU03.
- Se desglosan las historias de usuario en tareas, asignándoles un esfuerzo en horas, como se puede ver en el punto 6.1.1.1.

##### 6.1.1.1. Tareas planificadas

A continuación se enuncian las historias de usuario planificadas para el *Sprint*, su desglose en tareas y el esfuerzo calculado para las mismas:

- **HU01** Como usuario necesito visualizar la información de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Máxima]. Esta historia

de usuario se ha desglosado en las siguientes subtareas:

- Desarrollar, probar y publicar un servicio REST para consultar los datos de un contrato de obligación [8 horas]
- Desarrollar y probar una pantalla para consultar los datos de un contrato de obligación [8 horas]
- **HU02** Como usuario necesito visualizar la información las mejoras de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Alta]. Esta historia de usuario se ha desglosado en las siguientes subtareas:
  - Desarrollar, probar y publicar un servicio REST para consultar los datos mejoras de un contrato de obligación [8 horas]
  - Desarrollar y probar una pantalla para consultar los datos de mejoras de un contrato de obligación [8 horas]
- **HU03** Como usuario necesito visualizar la información de los expedientes de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Máxima].
  - Desarrollar, probar y publicar un servicio REST para consultar los datos de expedientes de un contrato de obligación [8 horas]
  - Desarrollar y probar una pantalla para consultar los datos de expedientes de un contrato de obligación [8 horas]

### 6.1.2. Pruebas de aceptación

**Nombre del test:** getContratoNotFound()

**DADO** que en el sistema no hay almacenado ningún contrato

**CUANDO** el usuario introduce el número de contrato "000005"

**ENTONCES** el sistema devolverá la siguiente información:

```
{'El contrato de obligación con id '000005' no existe',  
com.grupogimeno.siccagest.contratos.exceptions.ContratoNotFoundException,  
HttpStatus.NOT_FOUND)
```

**Nombre del test:** getContratoNotFound()

**DADO** que en el sistema hay almacenados los siguientes contratos:

ContratoOb 000002 (en el anexo B se encuentra la información completa en formato JSON)

**CUANDO** el usuario introduce el número de contrato "000005"

**ENTONCES** el sistema devolverá la siguiente información:

```
{'El contrato de obligación con id '000005' no existe',  
com.grupogimeno.siccagest.contratos.exceptions.ContratoNotFoundException,  
HttpStatus.NOT_FOUND)
```

**Nombre del test:** getContrato1()

**DADO** que en el sistema hay almacenados los siguientes contratos:

ContratoOb 000002 (en el Anexo B se encuentra la información anexo en formato JSON)

**CUANDO** el usuario introduce el número de contrato "000002"

**ENTONCES** el sistema devolverá la siguiente información:

ContratoOb 000002 (en el anexo B se encuentra la información completa en formato JSON)

### 6.1.3. Pruebas unitarias

La historia de usuario referente a la consulta de expedientes del contrato de obligación implica un filtrado de expedientes por las fechas introducidas. Este cálculo ha merecido una batería de tests unitarios:

**Nombre del test:** getExpedientesContratoNull()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=null, idExpedienteContrato=, fechaDesde=2017-03-01, fechaHasta=2017-09-30

**Salida esperada:** []

**Nombre del test:** getExpedientesFechaDesdeNull()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/03,fechaFin=31/08], idExpedienteContrato=, fechaDesde=2017-03-01, fechaHasta=null

**Salida esperada:** []

**Nombre del test:** getExpedientesFechaHastaNull()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/03,fechaFin=31/08], idExpedienteContrato=, fechaDesde=null, fechaHasta=2017-09-30

**Salida esperada:** []

**Nombre del test:** getExpedientesMismoAnyoDesdeYHastaDentroDeRango()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/03,fechaFin=31/08 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2017-03-01, fechaHasta=2017-08-31

**Salida esperada:** expedientes:[[idExpediente:001]]

**Nombre del test:** getExpedientesMismoAnyoDesdeDentroDeRango()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/03,fechaFin=31/08 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2017-03-01, fechaHasta=2017-08-31

**Salida esperada:** expedientes:[[idExpediente:001]]

**Nombre del test:** getExpedientesMismoAnyoHastaDentroDeRango()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/03,fechaFin=31/08 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2017-02-28, fechaHasta=2017-08-01

**Salida esperada:** expedientes:[[idExpediente:001]]

**Nombre del test:** getExpedientesMismoAnyoDesdeYHastaFueraDeRango()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/03,fechaFin=31/08 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2017-02-28, fechaHasta=2017-09-01

**Salida esperada:** expedientes:[[idExpediente:001]]

**Nombre del test:** getExpedientesMismoAnyoDesdeYHastaMayoresDeHastaRango()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/03,fechaFin=31/08 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2017-09-01, fechaHasta=2017-09-02

**Salida esperada:** []

**Nombre del test:** getExpedientesMismoAnyoDesdeYHastaMenoresDeDesdeRango()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/03,fechaFin=31/08 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2017-02-27, fechaHasta=2017-02-28

**Salida esperada:** []

**Nombre del test:** getExpedientesDistintoAnyoDesdeYHastaDentroDeRango()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/09,fechaFin=31/03 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2017-09-01, fechaHasta=2017-10-31

**Salida esperada:** expedientes:[[idExpediente:001]]

**Nombre del test:** getExpedientesDistintoAnyoDesdeYHastaMayoresDeHastaRango()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/09,fechaFin=31/03 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2017-04-01, fechaHasta=2017-04-02

**Salida esperada:** []

**Nombre del test:** getExpedientesDistintoAnyoDesdeYHastaMenoresDeDesdeRango()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/09,fechaFin=31/03 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2017-08-30, fechaHasta=2017-08-31

**Salida esperada:** []

**Nombre del test:** getExpedientesEnUnRangoDeDosAnyos()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/04,fechaFin=01/06 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2015-08-01, fechaHasta=2017-10-31

**Salida esperada:** []

**Nombre del test:** getExpedientesYFechasDistintoAnyoDesdeYHastaDentroDeRango()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/09,fechaFin=31/03 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2018-02-01, fechaHasta=2018-03-31

**Salida esperada:** expedientes:[[idExpediente:001]]

**Nombre del test:** getExpedientesYFechasDistintoAnyoDesdeDentroDeRango()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/09,fechaFin=31/03 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2017-09-01, fechaHasta=2018-04-01  
**Salida esperada:** expedientes:[[idExpediente:001]]

**Nombre del test:** getExpedientesYFechasDistintoAnyoHastaDentroDeRango()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/09,fechaFin=31/03 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2017-08-31, fechaHasta=2018-03-31

**Salida esperada:** expedientes:[[idExpediente:001]]

**Nombre del test:** getExpedientesYFechasDistintoAnyoDesdeYHastaFueraDeRango()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/09,fechaFin=31/03 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2017-08-31, fechaHasta=2018-04-01

**Salida esperada:** expedientes:[[idExpediente:001]]

**Nombre del test:** getExpedientesYFechasDistintoAnyoDesdeYHastaMayoresDeHastaRango()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/09,fechaFin=31/03 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2017-04-01, fechaHasta=2017-05-02

**Salida esperada:** []

**Nombre del test:** getExpedientesYFechasDistintoAnyoDesdeYHastaMenoresDeDesdeRango()

**Clase y método probados:** ContratoBuilder.getExpedientesContrato

**Parámetros de entrada:** contrato=[fechaIni=01/09,fechaFin=31/03 expedientes:[[idExpediente:001]], idExpedienteContrato=, fechaDesde=2017-07-30, fechaHasta=2017-08-31

**Salida esperada:** []

#### 6.1.4. Resultados obtenidos

Completamos totalmente las 3 primeras subtarefas de las 3 historias de usuario. Por la complejidad que tenia la primera pantalla a desarrollar, no pudimos acabarla en el tiempo previsto (suponía usar por primera vez Angular y TypeScript), pero dejamos las 3 subtarefas de frontend desarrolladas a falta de maquetar adecuadamente las pantallas.

En la figura 6.1 se puede ver el estado de las tareas asignadas al *Sprint* al finalizar el mismo.

Al final del *Sprint* disponemos de un *Incremento* del producto que se muestra en la figura 6.2

## 6.2. *Sprint* 2

El segundo *Sprint* se lleva a cabo en la quincena del 19-03-2018 al 30-03-2018.

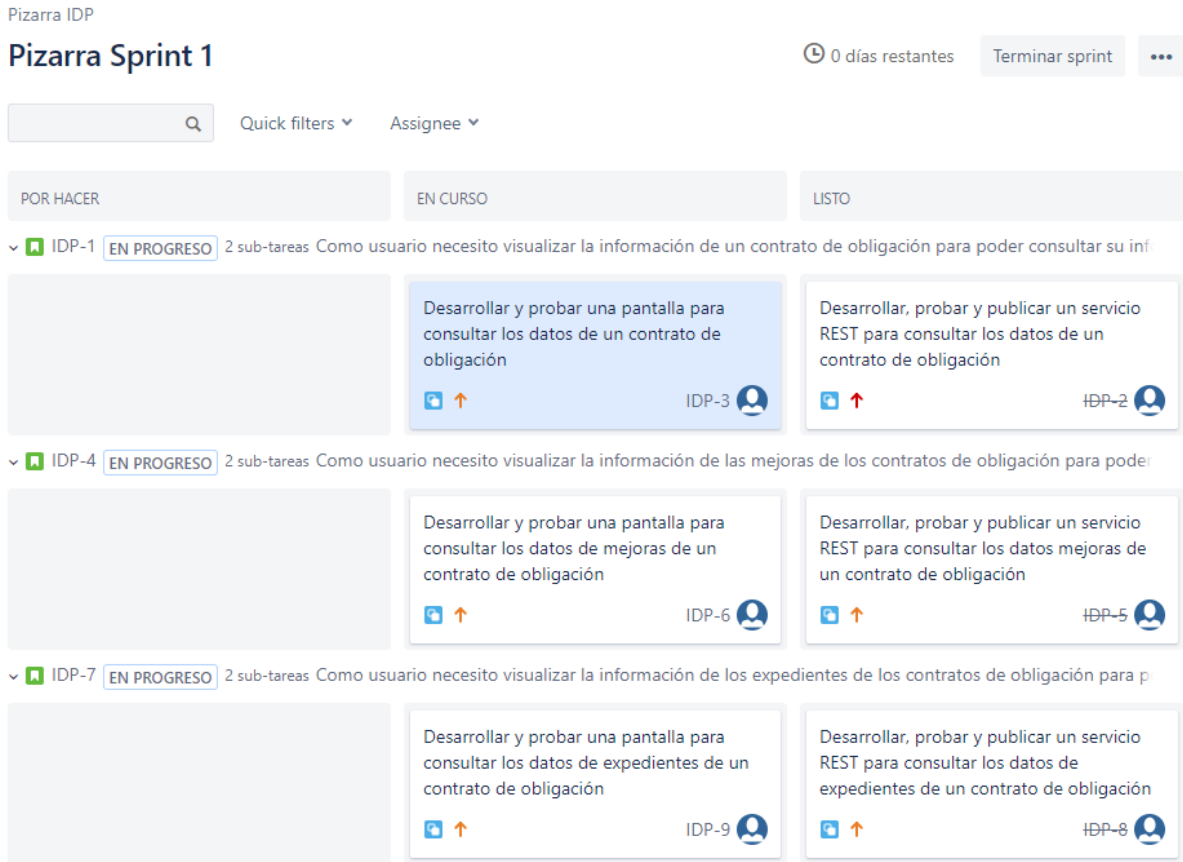


Figura 6.1: Estado de las tareas asignadas al *Sprint* 1 al finalizar dicho *Sprint*.

### 6.2.1. Planificación

En la reunión de planificación del *Spring* 2 participan la figura del *Scrum Master* (representada por el Supervisor del TFG), el *Product Owner* (representada por la Responsable de Software de ADC) y el equipo desarrollador (representado por Josep Ferrer).

En ella se toman decisiones críticas para el desarrollo del *Sprint*. Las más importantes, en este caso, son:

- El *Product Owner* acepta el prototipo de la figura 4.9 como el resultado visual que desea.
- Se modifica la pila del Producto como se puede ver en el punto 6.2.1.1.
- Todos los asistentes consensúan las Historias de usuario a incluir en el *Sprint*: HU04 y HU05.
- Se desglosan las historias de usuario en tareas, asignándoles un esfuerzo en horas, como se puede ver en el punto 6.2.1.2.

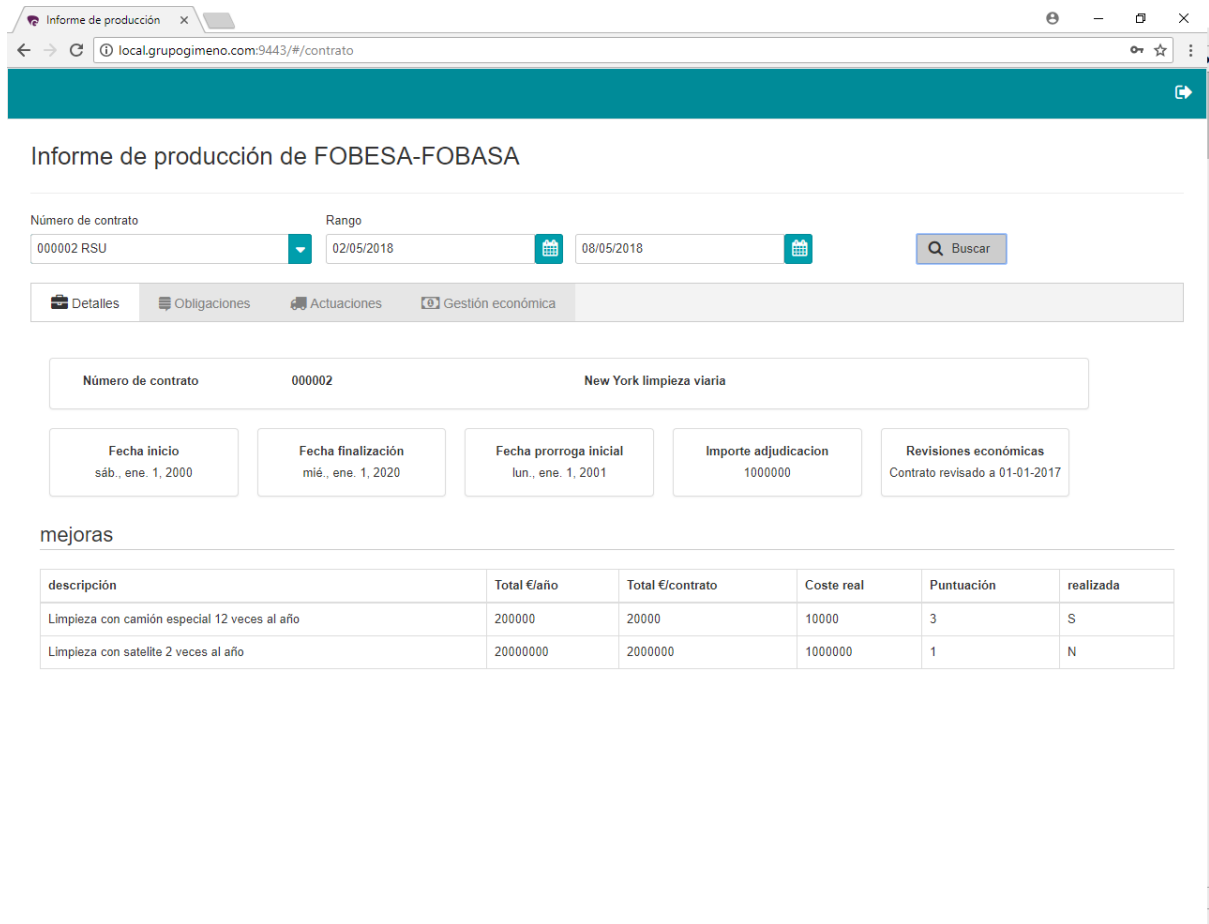


Figura 6.2: *Incremento* del producto tras *Sprint 1*.

#### 6.2.1.1. Actualización de la pila del producto

La pila del producto una vez actualizada ha quedado como sigue. Las historias de usuario en

- **verde** han quedado completadas.
  - **rojo** están pendientes de implementar.
  - **azul** están parcialmente completadas.
  - **magenta** las añadidas o modificadas.
  - **magenta** las eliminadas.
- 
- **HU01** Como usuario necesito visualizar la información de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Máxima].
  - **HU02** Como usuario necesito visualizar la información las mejoras de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Alta].

- **HU03** Como usuario necesito visualizar la información de los expedientes de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Máxima].
- **HU04** Como usuario necesito visualizar los pesajes de residuos por tipo de residuo relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU05** Como usuario necesito visualizar los vaciados de contenedores de residuos por fecha relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU06** Como usuario necesito visualizar los datos de gestión económica relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU07** Como usuario necesito visualizar la información de las horas comprometidas en los expedientes de contratos de obligación para poder consultar su información [48 puntos de historia] [Prioridad Media].
- **HU08** Como usuario necesito visualizar la información de las horas comprometidas y cumplidas en los expedientes de contratos de obligación para poder consultar su información [48 puntos de historia] [Prioridad Media].
- **HU09** Como usuario necesito visualizar el número de usuarios que han hecho uso del eco-parque relacionado con el contrato de obligación para poder consultar su información [16 puntos de historia] [Prioridad Baja].

#### 6.2.1.2. Tareas planificadas

Inicialmente añadimos las tareas no completadas en el *Sprint* 1:

- Desarrollar y probar una pantalla para consultar los datos de un contrato de obligación [8 horas]
- Desarrollar y probar una pantalla para consultar los datos de mejoras de un contrato de obligación [8 horas]
- Desarrollar y probar una pantalla para consultar los datos de expedientes de un contrato de obligación [8 horas]

A continuación se enuncian las historias de usuario planificadas para el *Sprint*, su desglose en tareas y el esfuerzo calculado para las mismas:

- **HU04** Como usuario necesito visualizar los pesajes de residuos por tipo de residuo relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media]. Esta historia de usuario se ha desglosado en las siguientes subtareas:



- Desarrollar y publicar un servicio REST para consultar los pesajes de residuos por tipo de residuo relacionados con un contrato de obligación [8 horas]
  - Desarrollar pruebas unitarias del servicio REST para consultar los pesajes de residuos por tipo de residuo relacionados con un contrato de obligación [8 horas]
  - Desarrollar y probar una pantalla para consultar pesajes de residuos por tipo de residuo relacionados con un contrato de obligación [8 horas]
- **HU05** Como usuario necesito visualizar los vaciados de contenedores de residuos por fecha relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media]. Esta historia de usuario se ha desglosado en las siguientes subtarefas:
- Desarrollar y publicar un servicio REST para consultar los contenedores descargados por tipo de residuo y fecha de descarga relacionados con un contrato de obligación [8 horas]
  - Desarrollar pruebas unitarias del servicio REST para consultar los los contenedores descargados por tipo de residuo y fecha de descarga relacionados con un contrato de obligación [8 horas]
  - Desarrollar y probar una pantalla para consultar los contenedores descargados por tipo de residuo y fecha de descarga relacionados con un contrato de obligación [8 horas]

Aunque inicialmente estaba previsto desarrollar otras historias de usuario, cambiamos la prioridad. La razón es que nos acababan de entregar los servicios que capturan estos datos desde los sensores. Era preferible probarlos por si surgía algún problema con estos nuevos servicios.

En la figura 6.3 se puede ver el estado completo de la pila del producto tras la planificación del *Sprint*.

### 6.2.2. Pruebas de aceptación

Todos los orígenes de datos de este *Sprint* son servicios REST ya existentes. Dichos servicios ya disponen de pruebas de aceptación. Esto hace que no sean necesario desarrollar pruebas de aceptación sobre ellos.

### 6.2.3. Pruebas unitarias

Todos los orígenes de datos de este *Sprint* son servicios REST ya existentes. Dichos servicios ya disponen de pruebas unitarias. Esto hace que no sea necesario desarrollar pruebas unitarias sobre ellos.

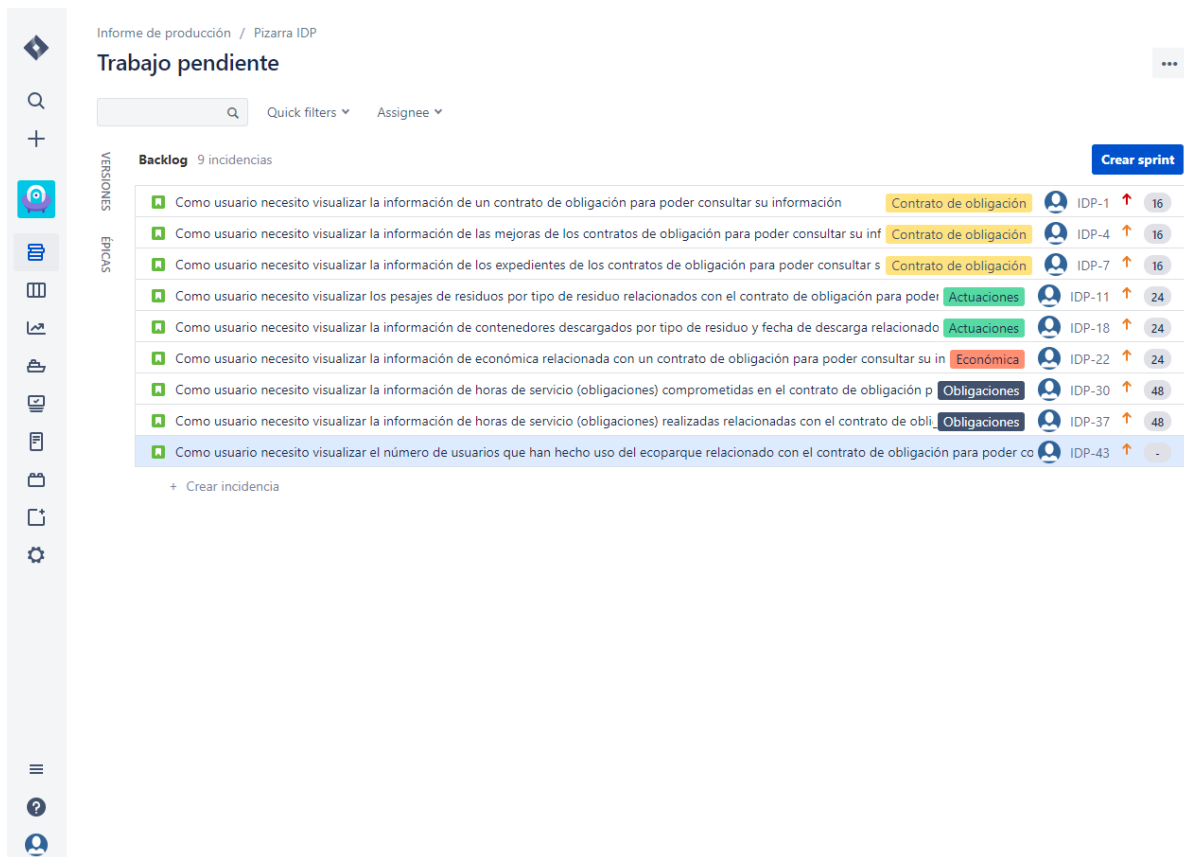


Figura 6.3: Product backlog tras planificar el *Sprint* 2.

#### 6.2.4. Resultados obtenidos

Las pruebas unitarias han supuesto más tiempo del esperado porque se trataba de una extensa casuística.

Completamos totalmente las tareas pendientes del primer *Sprint* y las 2 historias de usuario del *Sprint* 2.

En la figura 6.4 se puede ver el estado de las tareas asignadas al *Sprint* al finalizar el mismo:

Al final del *Sprint* disponemos de un *Incremento* del producto que se muestra en las figuras 6.5, 6.6 y 6.7

<div> <div>🕒 0 días restantes</div> <div>Terminar sprint</div> <div>⋮</div> </div>	
LISTO	
Desarrollar, probar y publicar un servicio REST para consultar los datos de un contrato de obligación	100-4
Desarrollar y probar una pantalla para consultar los datos de un contrato de obligación	100-4
Desarrollar, probar y publicar un servicio REST para consultar los datos mejoras de un contrato de obligación	100-4
Desarrollar y probar una pantalla para consultar los datos de mejoras de un contrato de obligación	100-4
Desarrollar, probar y publicar un servicio REST para consultar los datos de expedientes de un contrato de obligación	100-4
Desarrollar y probar una pantalla para consultar los datos de expedientes de un contrato de obligación	100-4
Desarrollar y publicar un servicio REST para consultar los pesajes de residuos por tipo de residuo relacionados con un contrato de obligación	100-14
Desarrollar pruebas unitarias del servicio REST para consultar los pesajes de residuos por tipo de residuo relacionados con un contrato de obligación	100-14
Desarrollar y probar una pantalla para consultar pesajes de residuos por tipo de residuo relacionados con un contrato de obligación	100-14
Desarrollar y publicar un servicio REST para consultar los contenedores descargados por tipo de residuo y fecha de descarga relacionados con un contrato de obligación	100-14
Desarrollar pruebas unitarias del servicio REST para consultar los los contenedores descargados por tipo de residuo y fecha de descarga relacionados con un contrato de obligación	100-20
Desarrollar y probar una pantalla para consultar los contenedores descargados por tipo de residuo y fecha de descarga relacionados con un contrato de obligación	100-20

Figura 6.4: Estado de las tareas asignadas al *Sprint* 2 al finalizar dicho *Sprint*.

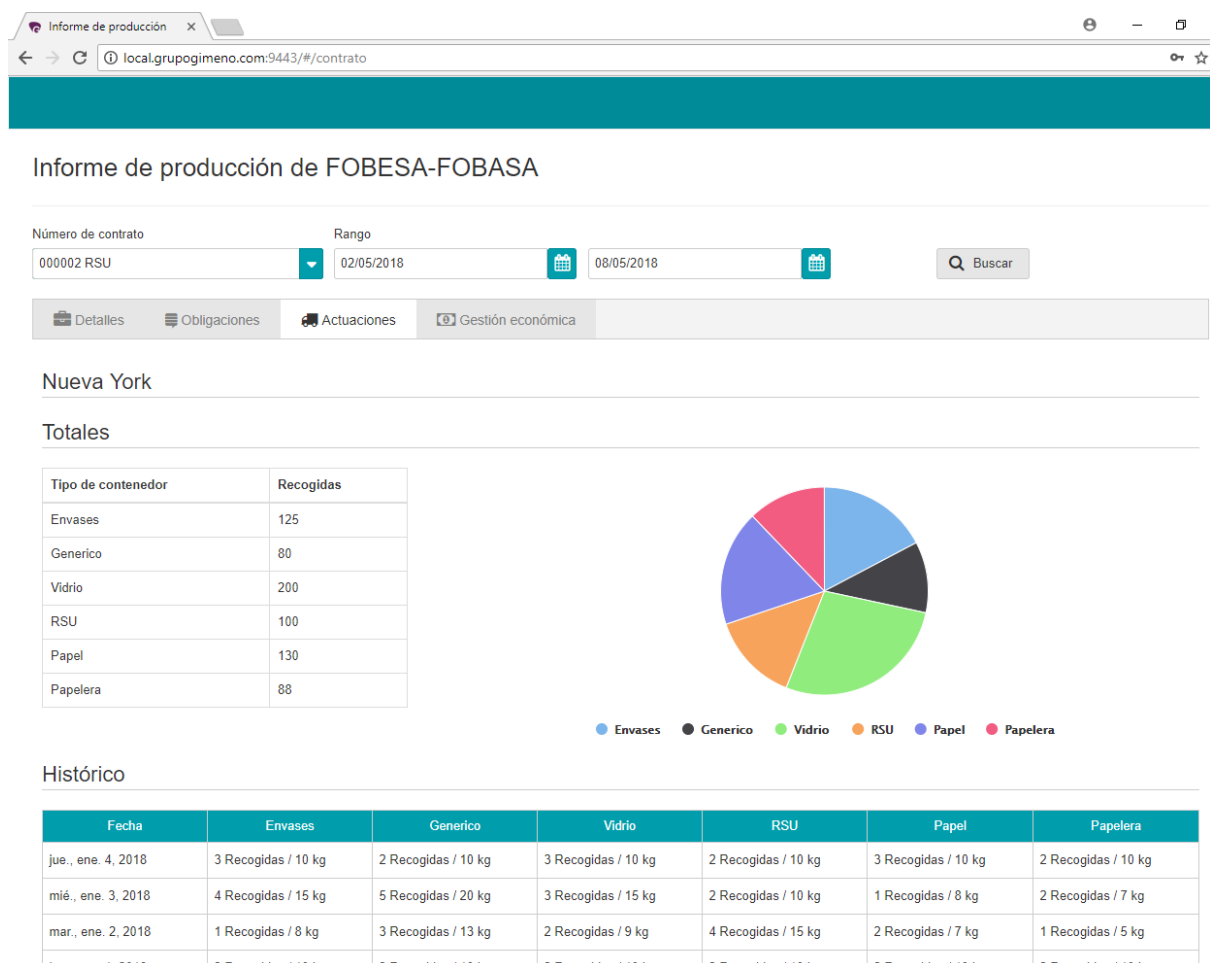


Figura 6.5: *Incremento del producto tras Sprint 2 (Región superior).*

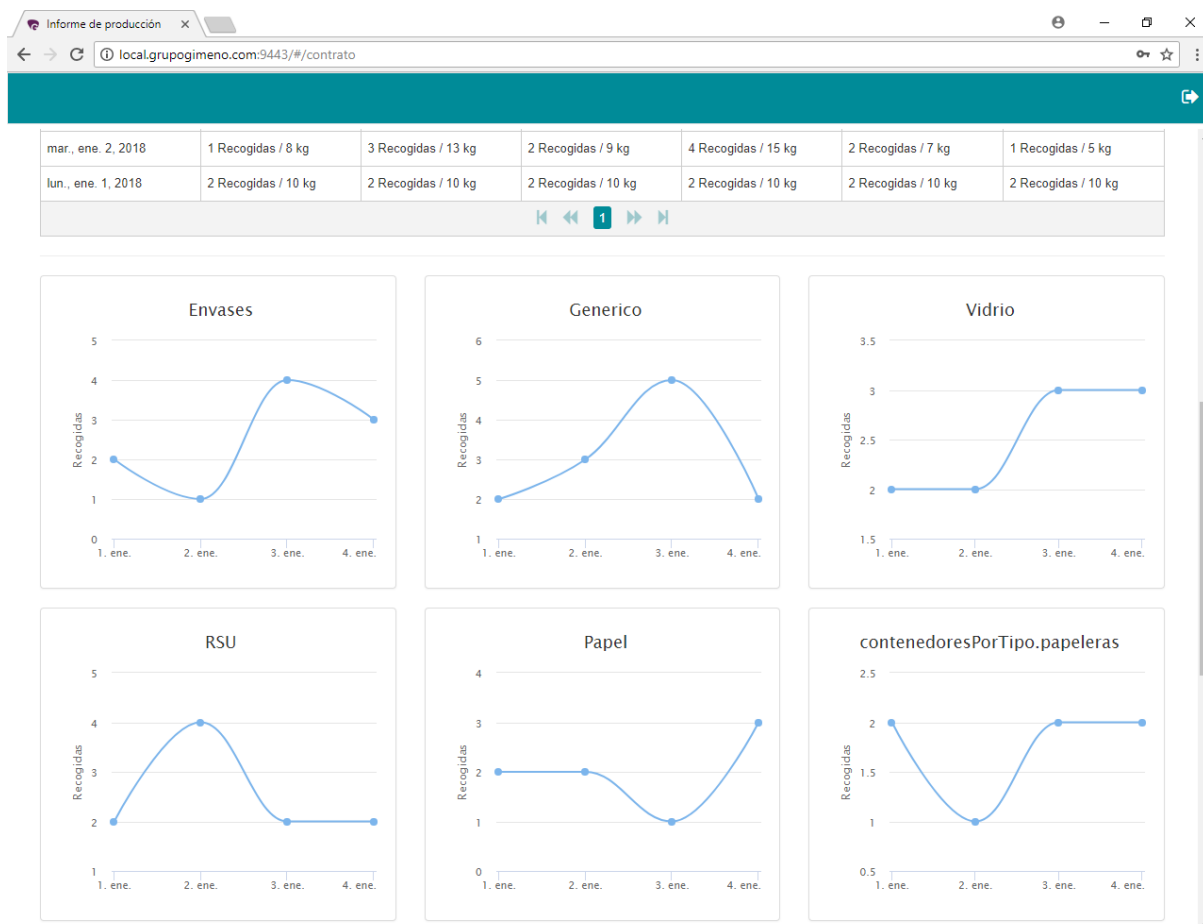


Figura 6.6: *Incremento del producto tras Sprint 2 (Región intermedia).*

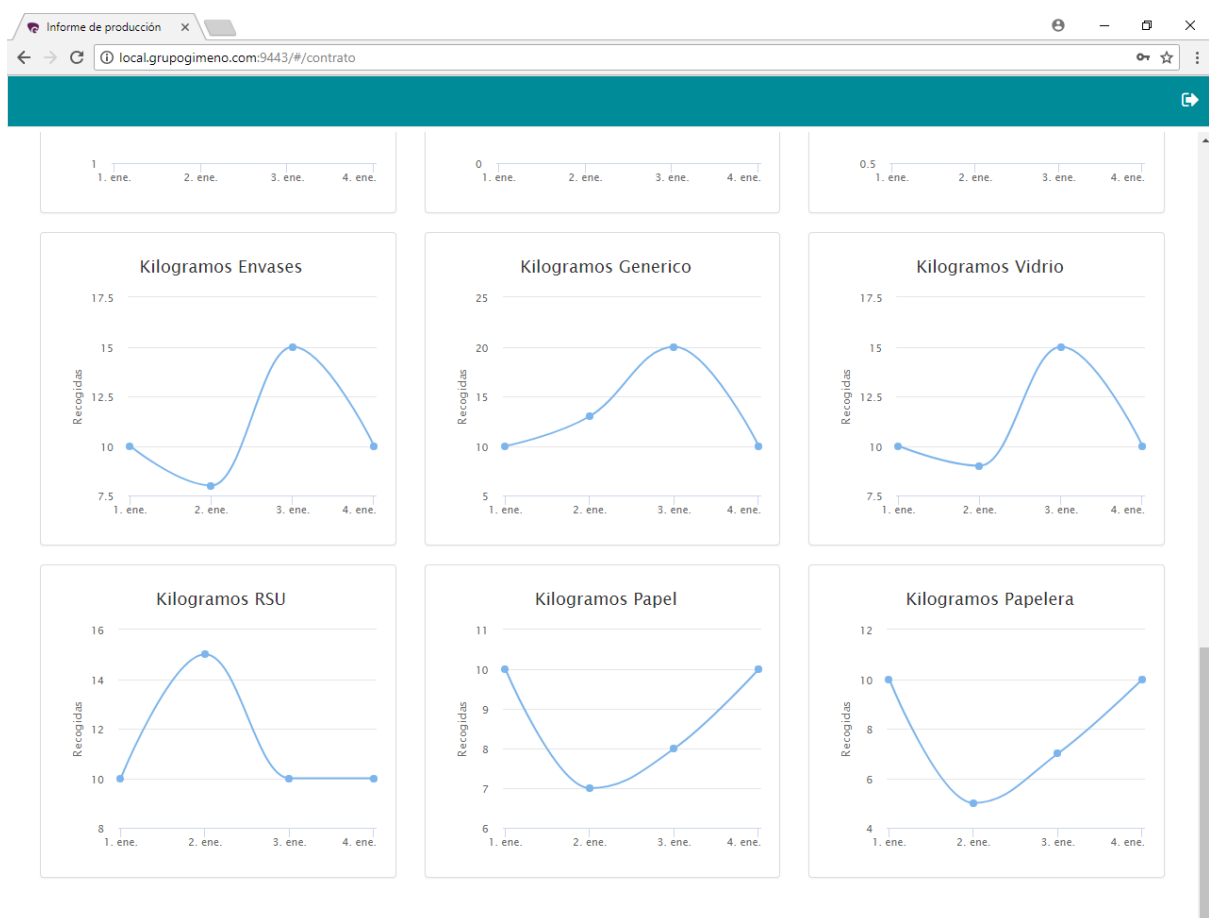


Figura 6.7: *Incremento del producto tras Sprint 2 (Región inferior).*

## 6.3. *Sprint* 3

El tercer *Sprint* se lleva a cabo en la quincena del 02-04-2018 al 13-04-2018.

### 6.3.1. Planificación

En la reunión de planificación del *Sprint* 3 participan la figura del *Scrum Master* (representada por el Supervisor del TFG), el *Product Owner* (representada por la Responsable de Software de ADC) y el equipo desarrollador (representado por Josep Ferrer).


En ella se toman decisiones críticas para el desarrollo del *Sprint*. Las más importantes, en este caso, son:

- El *Product Owner* acepta el prototipo de la figura 4.10 como el resultado visual que desea.
- Se modifica la pila del Producto como se puede ver en el punto 6.3.1.1.
- Todos los asistentes consensúan las Historias de usuario a incluir en el *Sprint*: HU06.
- Se desglosan las historias de usuario en tareas, asignándoles un esfuerzo en horas, como se puede ver en el punto 6.3.1.2.

#### 6.3.1.1. Actualización de la pila del producto

La pila del producto una vez actualizada ha quedado como sigue. Las historias de usuario en

- verde han quedado completadas.
  - rojo están pendientes de implementar.
  - azul están parcialmente completadas.
  - magenta las añadidas o modificadas.
  - ~~magenta~~ las eliminadas.
- 
- HU01 Como usuario necesito visualizar la información de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Máxima].
  - HU02 Como usuario necesito visualizar la información las mejoras de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Alta].
  - HU03 Como usuario necesito visualizar la información de los expedientes de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Máxima].

- **HU04** Como usuario necesito visualizar los pesajes de residuos por tipo de residuo relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU05** Como usuario necesito visualizar los vaciados de contenedores de residuos por fecha relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU06** Como usuario necesito visualizar los datos de gestión económica relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU07** Como usuario necesito visualizar la información de las horas comprometidas en los expedientes de contratos de obligación para poder consultar su información [48 puntos de historia] [Prioridad Media].
- **HU08** Como usuario necesito visualizar la información de las horas comprometidas y cumplidas en los expedientes de contratos de obligación para poder consultar su información [48 puntos de historia] [Prioridad Media].
- ~~**HU09** Como usuario necesito visualizar el  de usuarios que han hecho uso del eco-parque relacionado con el contrato de obliga para poder consultar su informa [16 puntos de historia] [Prioridad Baja]. \*~~

\* La historia de usuario HU09 ha sido finalmente descartada por el cliente. La razón es no disponer de un origen de datos para alimentar esta información.

#### 6.3.1.2. Tareas Planificadas

A continuación se enuncian las historias de usuario planificadas para el *Sprint* y su desglose en tareas:

- **HU06** Como usuario necesito visualizar los datos de gestión económica relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media]. Esta historia de usuario se ha desglosado en las siguientes subtareas:
  - Desarrollar y publicar un servicio REST para consultar la información económica relacionada con un contrato de obligación [8 horas]
  - Desarrollar pruebas unitarias del servicio REST para consultar la información económica relacionada con un contrato de obligación [8 horas]
  - Desarrollar y probar una pantalla para consultar la información económica relacionada con un contrato de obligación [8 horas]

Para implementar esta historia de usuario necesitamos ampliar el conocimiento sobre QueryDSL y sobre JPA. Esto es debido a que en la primera de sus tareas se centra en consultar los datos sobre una base de datos SQLServer construyendo una consulta SQL por medio de JPA y



QueryDSL.

Además de esta historia de usuario, y como sugerencia del supervisor, hemos planificado varias tareas de mejora y refactorización de la implementación de las historias de usuario ya desarrolladas:

- Unificar el servicio REST de contratos y contenedores en un nuevo servicio que se centre en la entidad Report [8 horas]
- Mejorar el tabla que muestra los contenedores descargados por día unificando en una sola tabla las actuales 6 (envases, genérico, RSU, vidrio, papel y papelería) [8 horas]
- Incluir en la pantalla una gráfica de contenedores por tipo [4 horas]
- Incluir en la pantalla una gráfica de contenedores por día [4 horas]

En la figura 6.8 se puede ver el estado completo de la pila del producto tras la planificación del *Sprint*.



Figura 6.8: Product backlog tras planificar el *Sprint* 3.

### 6.3.2. Pruebas de aceptación

**Nombre del test:** `getInformacionEconomicaWithoutFilters()`

**DADO** que en el sistema tiene la siguiente información económica: []

**CUANDO** el usuario solicita la información económica con `filterBy=null`

**ENTONCES** el sistema devolverá la siguiente información:

{'El servicio debe llamarse con: `idDepartamento`, `idExpediente`, `FechaDesde` y `fechaHasta`.,  
`com.grupogimeno.senda.starters.web.exceptions.BadRequestException`,  
`HttpStatus.BAD_REQUEST`)

**Nombre del test:** getInformacionEconomicaWith1Linea()

**DADO** que en el sistema tiene la siguiente información económica:

[Linea:[IdDocumento=010000000002, IdLinea=001, Importe=10.5, IdCeCo=0000110010, IdClaseCoste=0060320000, Fecha=20181230] Linea:[IdDocumento=010000000003, IdLinea=001, Importe=10.5, IdCeCo=0000110010, IdClaseCoste=0060320000, Fecha=20181231]]

**CUANDO** el usuario solicita la información económica con idDepartamento=7008, idExpediente=001, fechaDesde=20181230000000, fechaHasta=20181230000000

**ENTONCES** el sistema devolverá la siguiente información:

[Linea:[IdDocumento=010000000002, IdLinea=001, Importe=10.5, IdCeCo=0000110010, IdClaseCoste=0060320000, Fecha=20181230]

**Nombre del test:** getInformacionEconomicaWith2Lineas()

**DADO** que en el sistema tiene la siguiente información económica:

[Linea:[IdDocumento=010000000002, IdLinea=001, Importe=10.5, IdCeCo=0000110010, IdClaseCoste=0060320000, Fecha=20181230] Linea:[IdDocumento=010000000003, IdLinea=001, Importe=10.5, IdCeCo=0000110010, IdClaseCoste=0060320000, Fecha=20181231]]

**CUANDO** el usuario solicita la información económica con idDepartamento=7008, idExpediente=001, fechaDesde=20181230000000, fechaHasta=20181231000000

**ENTONCES** el sistema devolverá la siguiente información:

[Linea:[IdDocumento=010000000002, IdLinea=001, Importe=10.5, IdCeCo=0000110010, IdClaseCoste=0060320000, Fecha=20181230] Linea:[IdDocumento=010000000003, IdLinea=001, Importe=10.5, IdCeCo=0000110010, IdClaseCoste=0060320000, Fecha=20181231]]

**Nombre del test:** getInformacionEconomicaWithNonexistentIdDepartamento()

**DADO** que en el sistema tiene la siguiente información económica:

[Linea:[IdDocumento=010000000002, IdLinea=001, Importe=10.5, IdCeCo=0000110010, IdClaseCoste=0060320000, Fecha=20181230] Linea:[IdDocumento=010000000003, IdLinea=001, Importe=10.5, IdCeCo=0000110010, IdClaseCoste=0060320000, Fecha=20181231]]

**CUANDO** el usuario solicita la información económica con idDepartamento=NO\_EXISTO, idExpediente=001, fechaDesde=20181230000000, fechaHasta=20181231000000

**ENTONCES** el sistema devolverá la siguiente información: []

**Nombre del test:** getInformacionEconomicaWithNonexistentIdExpediente()

**DADO** que en el sistema tiene la siguiente información económica:

[Linea:[IdDocumento=010000000002, IdLinea=001, Importe=10.5, IdCeCo=0000110010, IdClaseCoste=0060320000, Fecha=20181230] Linea:[IdDocumento=010000000003, IdLinea=001, Importe=10.5, IdCeCo=0000110010, IdClaseCoste=0060320000, Fecha=20181231]]

**CUANDO** el usuario solicita la información económica con idDepartamento=001, idExpediente=NO\_EXISTO, fechaDesde=20181230000000, fechaHasta=20181231000000

**ENTONCES** el sistema devolverá la siguiente información: []

**Nombre del test:** getInformacionEconomicaWithBadFechaDesde()

**DADO** que en el sistema tiene la siguiente información económica:

[Linea:[IdDocumento=010000000002, IdLinea=001, Importe=10.5, IdCeCo=0000110010, IdClaseCoste=0060320000, Fecha=20181230] Linea:[IdDocumento=010000000003, IdLinea=001, Importe=10.5, IdCeCo=0000110010, IdClaseCoste=0060320000, Fecha=20181231]]

**CUANDO** el usuario solicita la información económica con idDepartamento=001, idExpediente=001, fechaDesde=FECHA\_INCORRECTA, fechaHasta=20181231000000

**ENTONCES** el sistema devolverá la siguiente información: []

**Nombre del test:** getInformacionEconomicaWithBadFechahasta()

**DADO** que en el sistema tiene la siguiente información económica:

[Linea:[IdDocumento=010000000002, IdLinea=001, Importe=10.5, IdCeCo=0000110010, IdClaseCoste=0060320000, Fecha=20181230] Linea:[IdDocumento=010000000003, IdLinea=001, Importe=10.5, IdCeCo=0000110010, IdClaseCoste=0060320000, Fecha=20181231]]

**CUANDO** el usuario solicita la información económica con idDepartamento=001, idExpediente=001, fechaDesde=20181231000000, fechaHasta=FECHA\_INCORRECTA

**ENTONCES** el sistema devolverá la siguiente información: []

### 6.3.3. Pruebas unitarias

La historia de usuario referente a la consulta de la información económica del contrato de obligación implica el cálculo del total de dicha información por clase de coste. Este cálculo ha merecido una batería de tests unitarios:

**Nombre del test:** getInformacionEconomicaNullList()

**Clase y método probados:** EconomicaBuilder.getTotalsByIdClaseCoste

**Parámetros de entrada:** informacionesEconomicas=null

**Salida esperada:** []

**Nombre del test:** getInformacionEconomicaEmptyList()

**Clase y método probados:** EconomicaBuilder.getTotalsByIdClaseCoste

**Parámetros de entrada:** informacionesEconomicas=[]

**Salida esperada:** []

**Nombre del test:** getInformacionEconomica1Line()

**Clase y método probados:** EconomicaBuilder.getTotalsByIdClaseCoste

**Parámetros de entrada:** informacionesEconomicas=[[idClaseCoste='0068127000', descripcion='Dot. Amort. instalaciones técnicas M.A.', importe=10.5]]

**Salida esperada:** [[idClaseCoste='0068127000', descripcion='Dot. Amort. instalaciones técnicas M.A.', importeTotal=10.5]]

**Nombre del test:** getInformacionEconomica2Line()

**Clase y método probados:** EconomicaBuilder.getTotalsByIdClaseCoste

**Parámetros de entrada:** informacionesEconomicas=[[idClaseCoste='0068127000', descripcion='Dot. Amort. instalaciones técnicas M.A.', importe=10.5], [idClaseCoste='0068127000', descripcion='Dot. Amort. instalaciones técnicas M.A.', importe=10.5]]

**Salida esperada:** [[idClaseCoste='0068127000', descripcion='Dot. Amort. instalaciones técnicas M.A.', importeTotal=21]]

**Nombre del test:** getInformacionEconomica2ClasesCoste()

**Clase y método probados:** EconomicaBuilder.getTotalsByIdClaseCoste

**Parámetros de entrada:** informacionesEconomicas=[[idClaseCoste='0068127000', descripcion='Dot. Amort. instalaciones técnicas M.A.', importe=10.5], [idClaseCoste='0000099301', descripcion='Generales Fobesa', importe=10.5]]

**Salida esperada:** [[idClaseCoste='0068127000', descripcion='Dot. Amort. instalaciones técnicas M.A.', importeTotal=10.5], [idClaseCoste='0068127000', descripcion='Generales Fobesa', importeTotal=10.5]]

**Nombre del test:** getInformacionEconomica2LineNegative()

**Clase y método probados:** EconomicaBuilder.getTotalsByIdClaseCoste

**Parámetros de entrada:** informacionesEconomicas=[[idClaseCoste='0068127000', descripcion='Dot. Amort. instalaciones técnicas M.A.', importe=10.5], [idClaseCoste='0068127000', descripcion='Dot. Amort. instalaciones técnicas M.A.', importe=-10.5]]

**Salida esperada:** [[idClaseCoste='0068127000', descripcion='Dot. Amort. instalaciones técnicas M.A.', importeTotal=0]]

#### 6.3.4. Resultados obtenidos

Completamos totalmente la historia de usuario y las tareas añadidas al *Sprint*.

En la figura 6.9 se puede ver el estado de las tareas asignadas al *Sprint* al finalizar el mismo:

Al final del *Sprint* disponemos de un *Incremento* del producto que se muestra en las figuras 6.10 y 6.11

### 6.4. *Sprint* 4

El cuarto *Sprint* se lleva a cabo en la quincena del 16-04-2018 al 27-04-2018

#### 6.4.1. Planificación

En la reunión de planificación del *Spring* 4 participan la figura del *Scrum Master* (representada por el Supervisor del TFG), el *Product Owner* (representada por la Responsable de Software de ADC) y el equipo desarrollador (representado por Josep Ferrer).

En ella se toman decisiones críticas para el desarrollo del *Sprint*. Las más importantes, en este caso, son:

- El *Product Owner* acepta el prototipo de la figura 4.8 como el resultado visual que desea.
- Se modifica la pila del Producto como se puede ver en el punto 6.4.1.1.
- Todos los asistentes consensúan las Historias de usuario a incluir en el *Sprint*: HU07.
- Se desglosan las historias de usuario en tareas, asignándoles un esfuerzo en horas, como se puede ver en el punto 6.4.1.2.

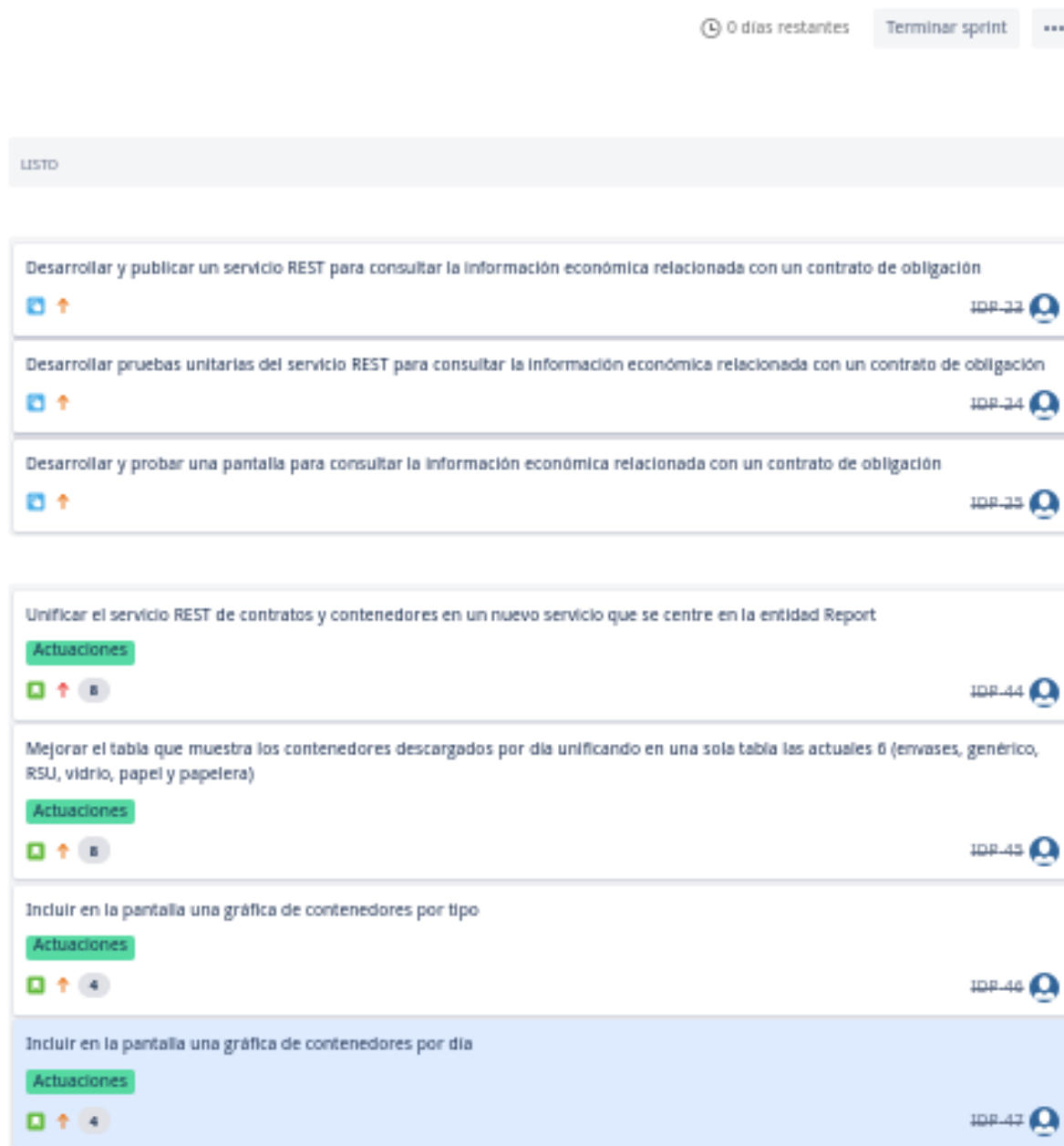


Figura 6.9: Estado de las tareas asignadas al *Sprint* 3 al finalizar dicho *Sprint*.

#### 6.4.1.1. Actualización de la pila del producto

La pila del producto una vez actualizada ha quedado como sigue. Las historias de usuario en

- verde han quedado completadas.
- rojo están pendientes de implementar.
- azul están parcialmente completadas.

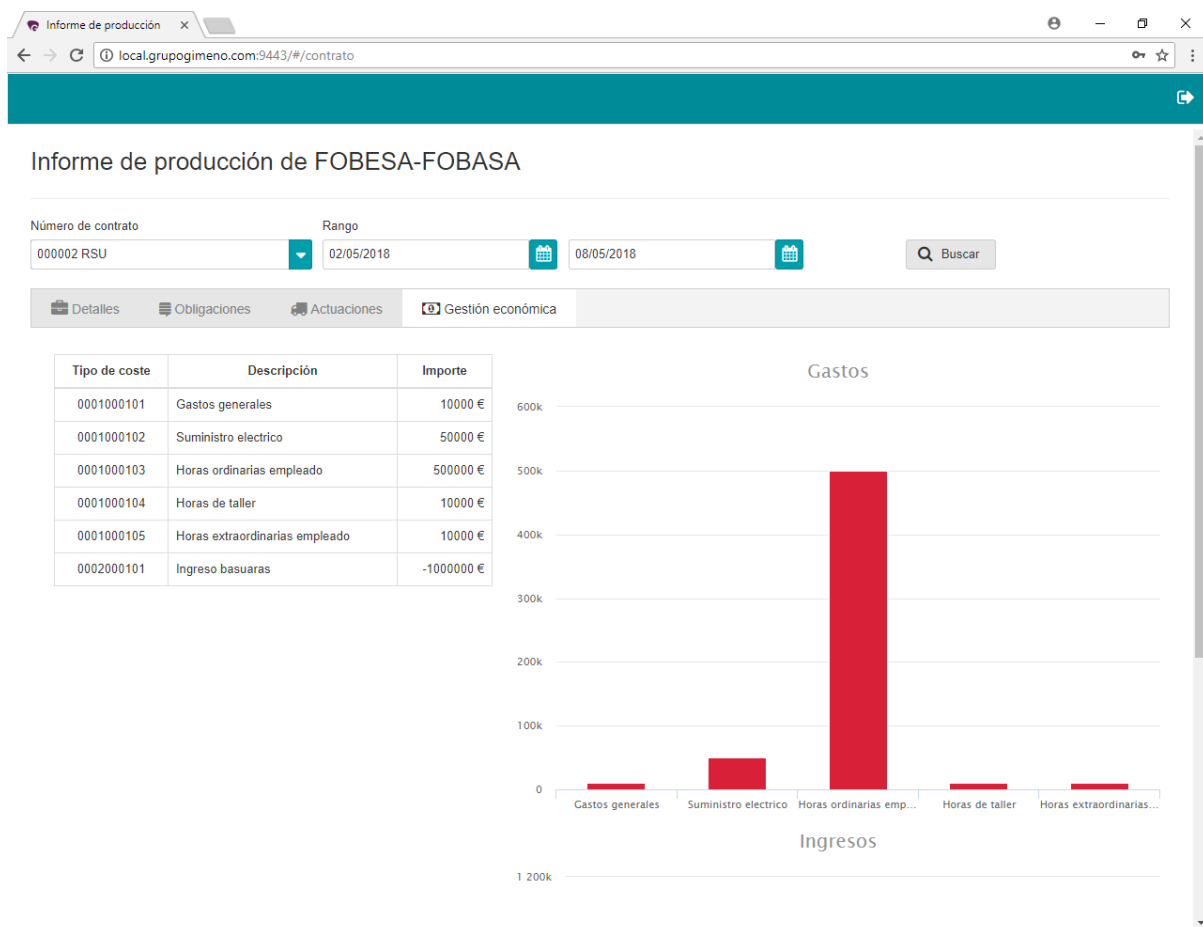


Figura 6.10: *Incremento del producto tras Sprint 3 (Región superior).*

- **magenta** las añadidas o modificadas.
- **magenta** las eliminadas.
- **HU01** Como usuario necesito visualizar la información de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Máxima].
- **HU02** Como usuario necesito visualizar la información las mejoras de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Alta].
- **HU03** Como usuario necesito visualizar la información de los expedientes de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Máxima].
- **HU04** Como usuario necesito visualizar los pesajes de residuos por tipo de residuo relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU05** Como usuario necesito visualizar los vaciados de contenedores de residuos por fecha relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].

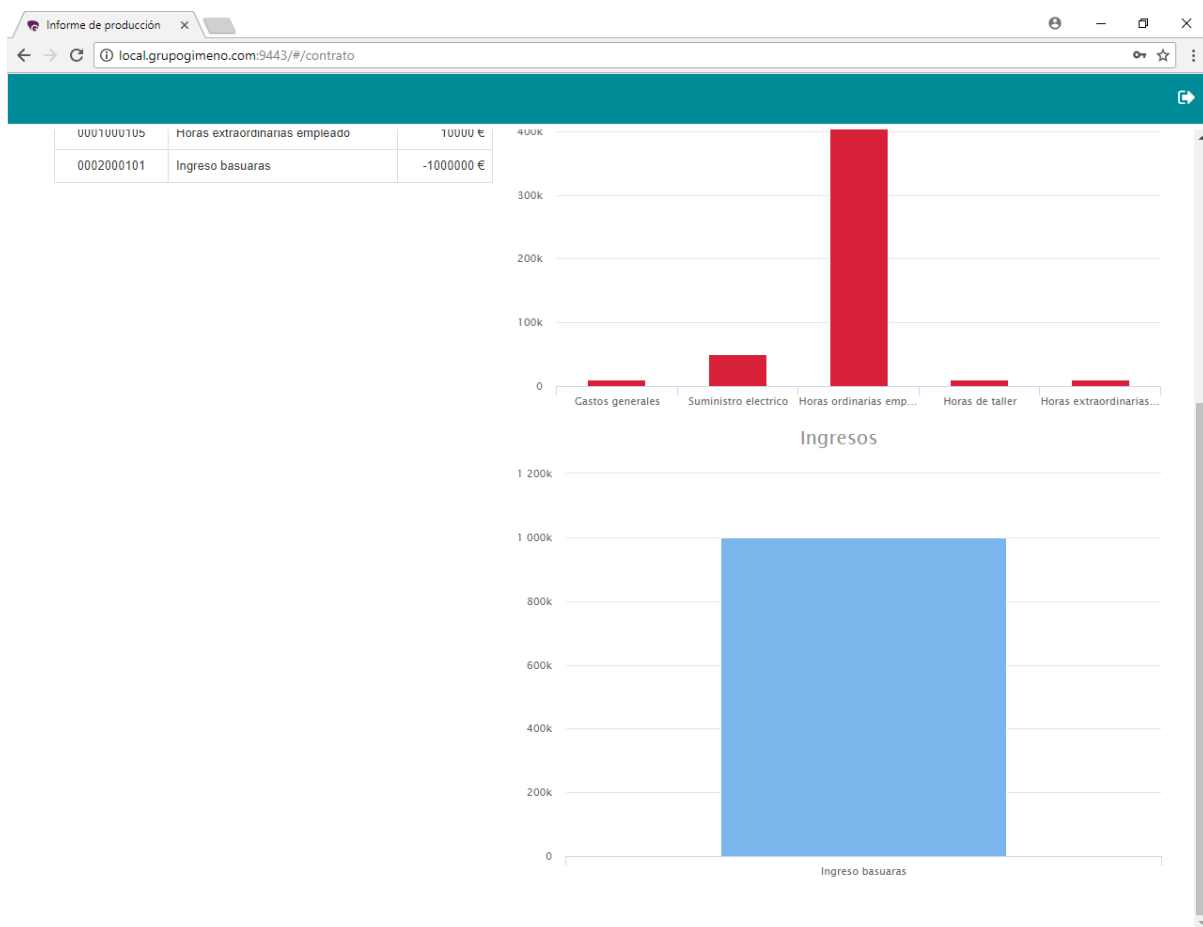


Figura 6.11: *Incremento del producto tras Sprint 3 (Región inferior).*

- **HU06** Como usuario necesito visualizar los datos de gestión económica relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU07** Como usuario necesito visualizar la información de las horas comprometidas en los expedientes de contratos de obligación para poder consultar su información [48 puntos de historia] [Prioridad Media].
- **HU08** Como usuario necesito visualizar la información de las horas comprometidas y cumplidas en los expedientes de contratos de obligación para poder consultar su información [48 puntos de historia] [Prioridad Media].
- ~~**HU09** Como usuario necesito visualizar el [ ] de usuarios que han hecho uso del eco par-~~  
~~que relacionado con el contrato de obliga para poder consultar su informa~~ [16 puntos de historia] [Prioridad Baja]. \*

\* La historia de usuario HU09 ha sido finalmente descartada por el cliente. La razón es no disponer de un origen de datos para alimentar esta información.

#### 6.4.1.2. Tareas planificadas

A continuación se enuncian las historias de usuario planificadas para el *Sprint* y su desglose en tareas:

- **HU07** Como usuario necesito visualizar la información de las horas comprometidas en los expedientes de contratos de obligación para poder consultar su información [48 puntos de historia] [Prioridad Media]. Esta historia de usuario se ha desglosado en las siguientes subtareas:
  - Desarrollar y publicar un servicio REST para consultar la información de obligaciones relacionada con un contrato de obligación [32 horas]
  - Desarrollar pruebas unitarias del servicio REST para consultar la información de obligaciones relacionada con un contrato de obligación [8 horas]
  - Desarrollar y probar una pantalla para consultar la información de obligaciones relacionada con un contrato de obligación [8 horas]

Para implementar esta historia de usuario necesitamos disponer de información sobre el origen de los datos y el tratamiento de su información para presentarla al usuario.

Además de esta historia de usuario, y como sugerencia del supervisor, hemos planificado varias tareas de mejora y refactorización de la implementación de las historias de usuario ya desarrolladas:

- Cambiar el servicio REST de información económica para hacerlo más general devolviendo la entidad completa de información económica [4 horas]
- Mejorar la presentación de contenedores con seis gráficos más (correspondientes a los kilogramos de envases, genérico, RSU, vidrio, papel y papelera) [4 horas]
- Mejorar la presentación económica con un diagrama de barras [4 horas]

En la figura 6.12 se puede ver el estado completo completo de la pila del producto tras la planificación del *Sprint*.

#### 6.4.2. Pruebas de aceptación

Las horas de servicio comprometidas se encuentran en la entidad Contrato de obligación. Las pruebas de aceptación correspondientes a las horas comprometidas están integradas en las pruebas de aceptación del Contrato de obligación.



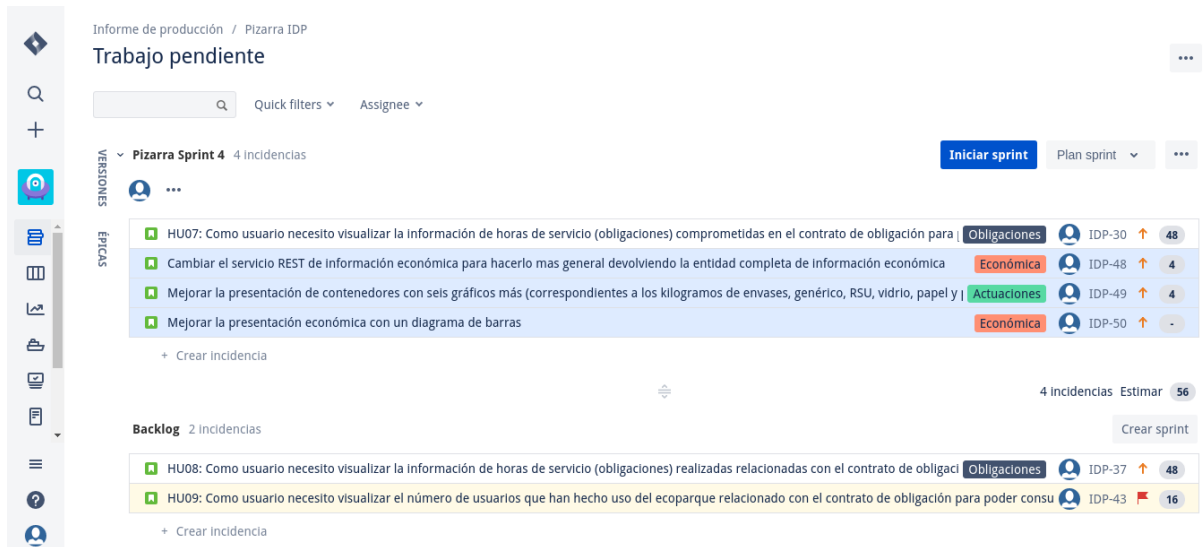


Figura 6.12: Product backlog tras planificar el *Sprint* 4.

### 6.4.3. Pruebas unitarias

La historia de usuario referente a la consulta de horas comprometidas del contrato de obligación implica un cálculo con una cierta complejidad. Este cálculo ha merecido una batería de tests unitarios:

**Nombre del test:** `getObligacionesEmptyList()`

**Clase y método probados:** `ReportBuidier.getObligacionesProgramadas`

**Parámetros de entrada:** `report=new Report()`, `partes=new HashSet()`, `fechaDesde=2018-01-01`, `fechaHasta=2018-02-01`

**Salida esperada:** `[]`

**Nombre del test:** `getObligaciones6Dias()`

**Clase y método probados:** `ReportBuidier.getObligacionesProgramadas`

**Parámetros de entrada:** `report:horasDiarias=6.0`, `lunes=true`, `martes=true`, `miercoles=true`, `jueves=true`, `viernes=true`, `sabado=true`, `domingo=false`, `partes:horas=1.0`, `fechaDesde=2018-01-01`, `fechaHasta=2018-01-02`

**Salida esperada:** `[horasProgramadas=6.0, horasRealizadas=3.0]`

**Nombre del test:** `getObligaciones2Semanas()`

**Clase y método probados:** `ReportBuidier.getObligacionesProgramadas`

**Parámetros de entrada:** `report:horasSemanales=7.0`, `partes:horas=1.0`, `fechaDesde=2018-01-01`, `fechaHasta=2018-01-02`

**Salida esperada:** `[horasProgramadas=1.0, horasRealizadas=3.0]`

### 6.4.4. Resultados obtenidos

Completamos totalmente la historia de usuario y las tareas añadidas al *Sprint*.

En la figura 6.13 se puede ver el estado de las tareas asignadas al *Sprint* al finalizar el mismo.

Al final del *Sprint* disponemos de un *Incremento* del producto que se muestra en la figura 6.14

## 6.5. *Sprint* 5

El quinto *Sprint* se lleva a cabo en la quincena del 30-04-2018 al 11-05-2018.

### 6.5.1. Planificación

En la reunión de planificación del *Sprint* 5 participan la figura del *Scrum Master* (representada por el Supervisor del TFG), el *Product Owner* (representada por la Responsable de Software de ADC) y el equipo desarrollador (representado por Josep Ferrer).


En ella se toman decisiones críticas para el desarrollo del *Sprint*. Las más importantes, en este caso, son:

- El *Product Owner* acepta el prototipo de la figura 4.8 como el resultado visual que desea.
- Se modifica la pila del Producto como se puede ver en el punto 6.5.1.1.
- Todos los asistentes consensúan las Historias de usuario a incluir en el *Sprint*: HU08.
- Se desglosan las historias de usuario en tareas, asignándoles un esfuerzo en horas, como se puede ver en el punto 6.5.1.2.

#### 6.5.1.1. Actualización de la pila del producto

La pila del producto una vez actualizada ha quedado como sigue. Las historias de usuario en

- verde han quedado completadas.
  - rojo están pendientes de implementar.
  - azul están parcialmente completadas.
  - magenta las añadidas o modificadas.
  - magenta las eliminadas.
- 
- HU01 Como usuario necesito visualizar la información de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Máxima].

- **HU02** Como usuario necesito visualizar la información las mejoras de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Alta].
- **HU03** Como usuario necesito visualizar la información de los expedientes de los contratos de obligación para poder consultar su información [16 puntos de historia] [Prioridad Máxima].
- **HU04** Como usuario necesito visualizar los pesajes de residuos por tipo de residuo relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU05** Como usuario necesito visualizar los vaciados de contenedores de residuos por fecha relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU06** Como usuario necesito visualizar los datos de gestión económica relacionados con el contrato de obligación para poder consultar su información [24 puntos de historia] [Prioridad Media].
- **HU07** Como usuario necesito visualizar la información de las horas comprometidas en los expedientes de contratos de obligación para poder consultar su información [48 puntos de historia] [Prioridad Media].
- **HU08** Como usuario necesito visualizar la información de las horas cumplidas en los expedientes de contratos de obligación para poder consultar su información [48 puntos de historia] [Prioridad Media].
- ~~**HU09** Como usuario necesito visualizar el  de usuarios que han hecho uso del eco-parque relacionado con el contrato de obliga para poder consultar su informa [16 puntos de historia] [Prioridad Baja]. \*~~

\* La historia de usuario HU09 ha sido finalmente descartada por el cliente. La razón es no disponer de un origen de datos para alimentar esta información.

#### 6.5.1.2. Tareas planificadas

A continuación se enuncian las historias de usuario planificadas para el *Sprint* y su desglose en tareas:

- **HU08** Como usuario necesito visualizar la información de las horas cumplidas en los expedientes de contratos de obligación para poder consultar su información [48 puntos de historia] [Prioridad Media]. Esta historia de usuario se ha desglosado en las siguientes subtareas:
  - Desarrollar y publicar un servicio REST para consultar la información de horas de obligaciones realizadas con un contrato de obligación [32 horas]
  - Desarrollar pruebas unitarias del servicio REST para consultar la información de obligaciones relacionadas relacionadas con un contrato de obligación [8 horas]

- Desarrollar y probar una pantalla para consultar la información de obligaciones relacionada con un contrato de obligación [8 horas]

Para implementar esta historia de usuario necesitamos disponer de información sobre el origen de los datos y el tratamiento de su información para presentarla al usuario. La empresa nos ha proporcionado el código fuente del programa actual. Está escrito en VisualBasic y tiene embebidas sentencias SQL nativas. Tras analizarlo llegamos a las siguientes conclusiones:

- El código VisualBasic es suficientemente inteligible por nuestra parte para que pueda servir como único origen de información.
- El algoritmo desarrollado se usa para dos informes diferentes. Es mucho más complejo que el algoritmo necesario para el informe de producción.

Como consecuencia decidimos usarlo como aproximación a fuente de información pero no puede servirnos como único origen de información.

Además de esta historia de usuario hemos planificado varias tareas de mejora y refactorización de la implementación de las historias de usuario ya desarrolladas:

- Incluir en pantalla los desplegables de Departamento, Cliente y Expediente. Esto supone construirlos con los datos recibidos del Contrato de obligación y recargarlos al seleccionar el contenido de uno de ellos [4 horas],
- Refactorizar los test para usar buidiers [2 horas].

En la figura 6.15 se puede ver el estado completo completo de la pila del producto tras la planificación del *Sprint*.

### 6.5.2. Pruebas de aceptación

**Nombre del test:** getPartesWithoutFilters()

**DADO** que en el sistema tiene la siguiente información de partes de trabajo: []

**CUANDO** el usuario solicita la información de partes de trabajo con filterBy=null

**ENTONCES** el sistema devolverá la siguiente información:

{'El servicio debe llamarse con: idDepartamento, idExpediente, FechaDesde y fechaHasta., com.grupogimeno.senda.starters.web.exceptions.BadRequestException, HttpStatus.BAD\_REQUEST)

**Nombre del test:** getPartesWith1Linea()

**DADO** que en el sistema tiene la siguiente información económica:

[ParteTrabajoCab: IdDepartamento=001, IdAnyoParte=2018, IdParte=000043, FFintrabajos=2018-01-01 00:00:00.00, [LineaHorasEmpleado:[IdDepartamento=001, IdAnyoParte=2018, IdParte=000043, LineaHEmp=1, IdCategoriaPlicas=3010, HorasNormales=10.0, HorasExtra=10.0], LineaHoras-Vehiculo:[IdDepartamento=001, IdAnyoParte=2018, IdParte=000043, LineaHVeh=1, IdVehiculo=CS8096AH, Horas=10.0],

[ParteTrabajoCab: IdDepartamento=001, IdAnyoParte=2018, IdParte=000044, FFintrabajos=2018-01-02 00:00:00.00, [LineaHorasEmpleado:[IdDepartamento=001, IdAnyoParte=2018, IdParte=000044, LineaHEmp=1, IdCategoriaPlicas=3010, HorasNormales=10.0, HorasExtra=10.0], LineaHoras-Vehiculo:[IdDepartamento=001, IdAnyoParte=2018, IdParte=000044, LineaHVeh=1, IdVehiculo=CS8096AH, Horas=10.0,]]

**CUANDO** el usuario solicita la información económica con idDepartamento=7008, idExpediente=001, fechaDesde=20180101000000, fechaHasta=20180101000000

**ENTONCES** el sistema devolverá la siguiente información:

[ParteTrabajoCab: IdDepartamento=001, IdAnyoParte=2018, IdParte=000043, FFintrabajos=2018-01-01 00:00:00.00, [LineaHorasEmpleado:[IdDepartamento=001, IdAnyoParte=2018, IdParte=000043, LineaHEmp=1, IdCategoriaPlicas=3010, HorasNormales=10.0, HorasExtra=10.0], LineaHoras-Vehiculo:[IdDepartamento=001, IdAnyoParte=2018, IdParte=000043, LineaHVeh=1, IdVehiculo=CS8096AH, Horas=10.0,]]

**Nombre del test:** getPartesWith2Lineas()

**DADO** que en el sistema tiene la siguiente información económica:

[ParteTrabajoCab: IdDepartamento=001, IdAnyoParte=2018, IdParte=000043, FFintrabajos=2018-01-01 00:00:00.00, [LineaHorasEmpleado:[IdDepartamento=001, IdAnyoParte=2018, IdParte=000043, LineaHEmp=1, IdCategoriaPlicas=3010, HorasNormales=10.0, HorasExtra=10.0], LineaHoras-Vehiculo:[IdDepartamento=001, IdAnyoParte=2018, IdParte=000043, LineaHVeh=1, IdVehiculo=CS8096AH, Horas=10.0,],

[ParteTrabajoCab: IdDepartamento=001, IdAnyoParte=2018, IdParte=000044, FFintrabajos=2018-01-02 00:00:00.00, [LineaHorasEmpleado:[IdDepartamento=001, IdAnyoParte=2018, IdParte=000044, LineaHEmp=1, IdCategoriaPlicas=3010, HorasNormales=10.0, HorasExtra=10.0], LineaHoras-Vehiculo:[IdDepartamento=001, IdAnyoParte=2018, IdParte=000044, LineaHVeh=1, IdVehiculo=CS8096AH, Horas=10.0,]]

**CUANDO** el usuario solicita la información económica con idDepartamento=7008, idExpediente=001, fechaDesde=20180101000000, fechaHasta=20180102000000

**ENTONCES** el sistema devolverá la siguiente información:

[ParteTrabajoCab: IdDepartamento=001, IdAnyoParte=2018, IdParte=000043, FFintrabajos=2018-01-01 00:00:00.00, [LineaHorasEmpleado:[IdDepartamento=001, IdAnyoParte=2018, IdParte=000043, LineaHEmp=1, IdCategoriaPlicas=3010, HorasNormales=10.0, HorasExtra=10.0], LineaHoras-Vehiculo:[IdDepartamento=001, IdAnyoParte=2018, IdParte=000043, LineaHVeh=1, IdVehiculo=CS8096AH, Horas=10.0,],

[ParteTrabajoCab: IdDepartamento=001, IdAnyoParte=2018, IdParte=000044, FFintrabajos=2018-01-02 00:00:00.00, [LineaHorasEmpleado:[IdDepartamento=001, IdAnyoParte=2018, IdParte=000044, LineaHEmp=1, IdCategoriaPlicas=3010, HorasNormales=10.0, HorasExtra=10.0], LineaHoras-Vehiculo:[IdDepartamento=001, IdAnyoParte=2018, IdParte=000044, LineaHVeh=1, IdVehiculo=CS8096AH, Horas=10.0,]]

### 6.5.3. Pruebas unitarias

Las pruebas unitarias de este *Sprint* suponen una modificación de las pruebas unitarias del *Sprint* anterior. Además de comprobar el total de horas programadas se comprueba el total de horas realizadas.

#### 6.5.4. Resultados obtenidos

Completamos totalmente la historia de usuario y las tareas añadidas al *Sprint*.

En la figura 6.16 se puede ver el estado de las tareas asignadas al *Sprint* al finalizar el mismo.

Al final del *Sprint* disponemos de un *Incremento* del producto que se muestra en las figura 6.17



Figura 6.13: Estado de las tareas asignadas al *Sprint* 4 al finalizar dicho *Sprint*.

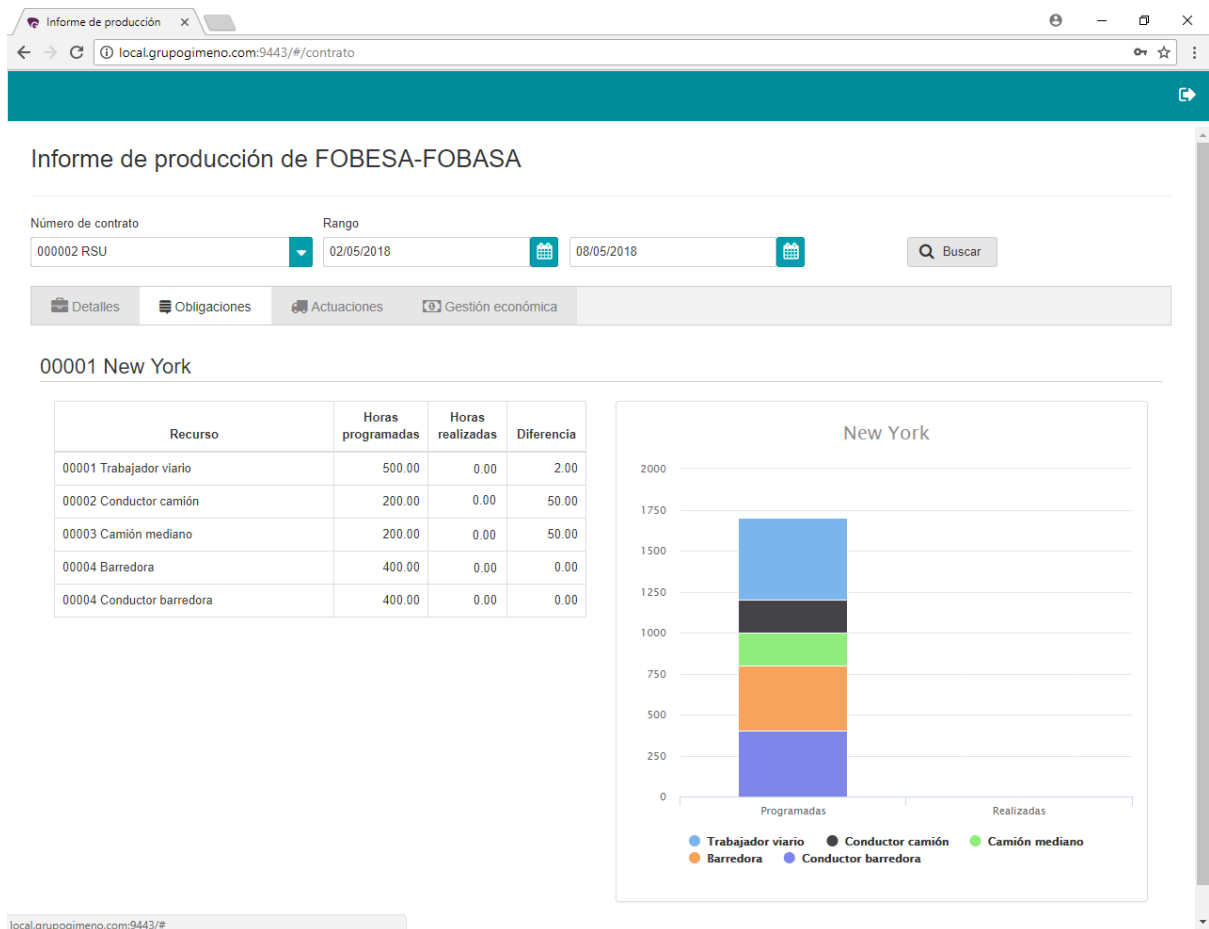


Figura 6.14: Incremento del producto tras *Sprint 4*.

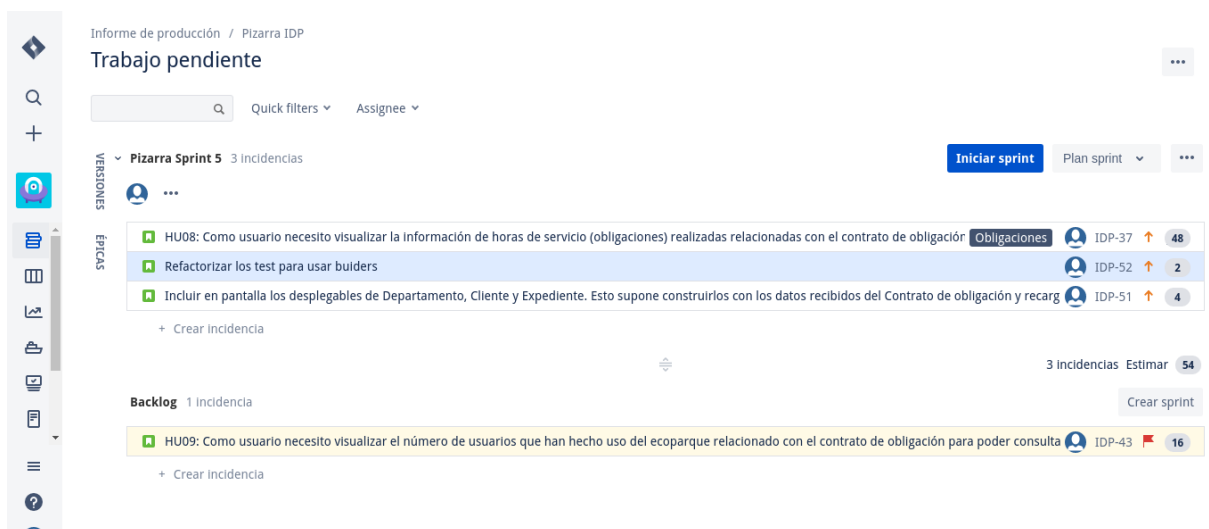


Figura 6.15: Product backlog tras planificar el *Sprint 5*.



🕒 0 días restantes

Terminar sprint

⋮

LISTO

a poder valorar su nivel de cumplimiento

Desarrollar y publicar un servicio REST para consultar la información de horas de obligaciones realizadas con un contrato de obligación

🔗 ⬆

IDP-38

👤

Desarrollar pruebas unitarias del servicio REST para consultar la información de obligaciones relacionadas relacionadas con un contrato de obligación

🔗 ⬆

IDP-39

👤

Desarrollar y probar una pantalla para consultar la información de obligaciones relacionada con un contrato de obligación

🔗 ⬆

IDP-40

👤

Refactorizar los test para usar buiders

✅ ⬆ 2

IDP-52

👤

Incluir en pantalla los desplegados de Departamento, Cliente y Expediente. Esto supone construirlos con los datos recibidos del Contrato de obligación y recargarlos al seleccionar el contenido de uno de ellos

✅ ⬆ 4

IDP-51

👤

Figura 6.16: Estado de las tareas asignadas al *Sprint* 5 al finalizar dicho *Sprint*.

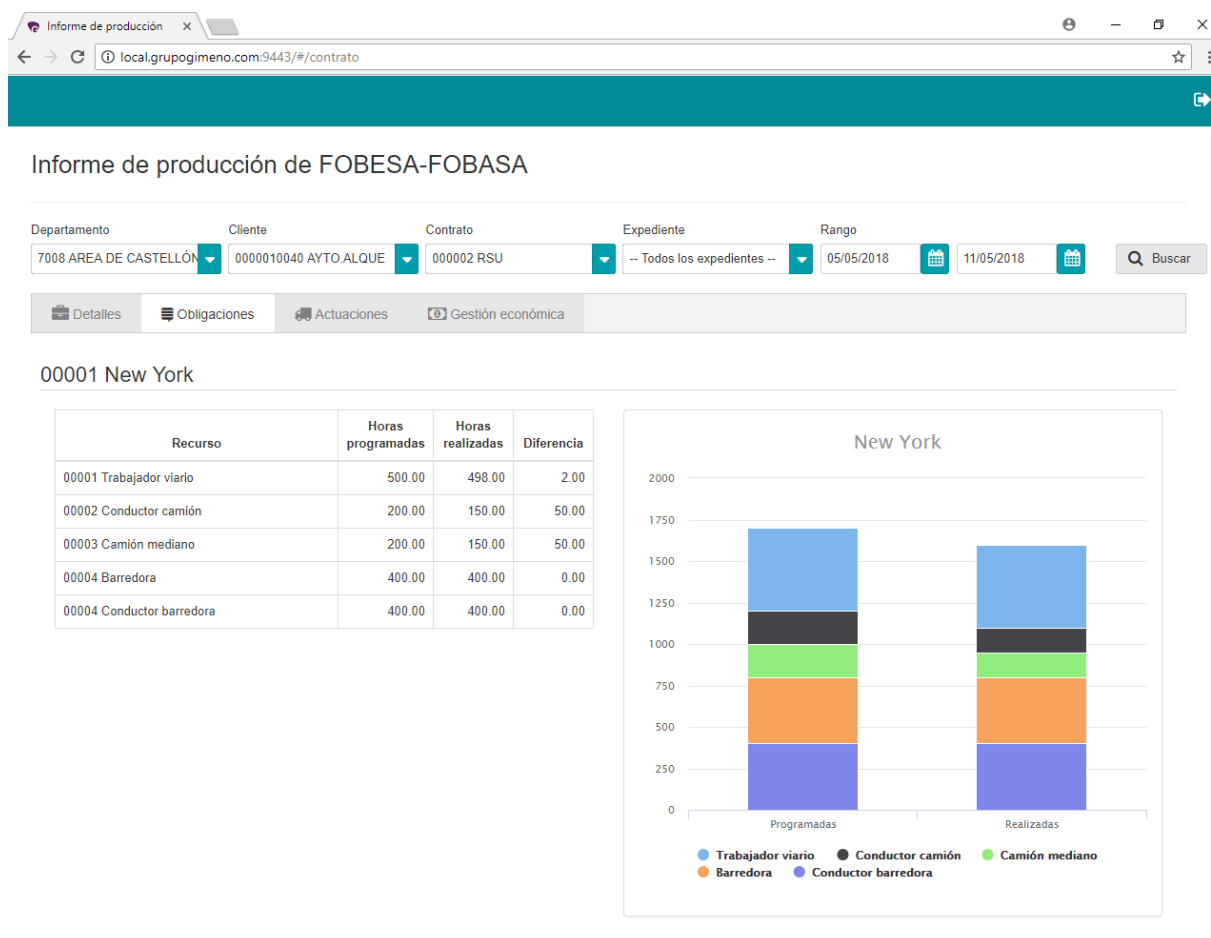


Figura 6.17: *Incremento del producto tras el Sprint 5.*

## Capítulo 7

# Conclusiones

Respecto los objetivos planteados al inicio del proyecto podemos afirmar que se han cumplido totalmente.

En el ámbito formativo y profesional ha sido muy interesante usar **Scrum** en un proyecto real. Igualmente ha sido muy enriquecedor el uso por primera vez de **Angular**, **TypeScript**, **JavaScript** y el protocolo **Git Flow**.

Como experiencia personal, no es la primera vez que estoy en un ámbito laboral pero si que es la primera vez que trabajo en un entorno tan actual e innovador.

A continuación hacemos referencia al futuro crecimiento de la aplicación y sus extensiones.

### 7.1. Modificaciones y futuras extensiones

La aplicación desarrollada se le presentará al cliente (FOBESA). Habrá que tener en cuenta el *feedback* recibido en la reunión con éste para realizar ajustes y modificaciones.

Una modificación que habrá que hacer será la integración en el portal de aplicaciones. El resto de modificaciones dependerá de la opinión del cliente.

Respecto a las futuras extensiones debemos mencionar que estarán abiertas a la necesidades del cliente final. A pesar de ello parece razonable que la principal extensión sea el uso de la aplicación por el cliente final de FOBESA (ayuntamiento, diputación o otras administraciones). Esto supondrá la creación de un Rol de usuario ClienteFinal. Para todos los usuarios con este rol deberán mostrarse solo las solapas de Contrato, Obligaciones y Actuaciones. Se dejará para uso interno de la empresa todo lo que se refiere a Información Económica.



# Bibliografía

- [1] Grupo gimeno. <http://www.grupogimeno.com>. [Consulta: 1 de Mayo de 2018].
- [2] Historias de usuario. <http://www.pmoinformatica.com/2013/04/que-son-las-historias-de-usuario-7.html>. [Consulta: 10 de Mayo de 2018].
- [3] Historias de usuario. [https://es.wikipedia.org/wiki/Scrum\\_\(desarrollo\\_de\\_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)). [Consulta: 10 de Mayo de 2018].
- [4] Introducing git flow. <https://datasift.github.io/gitflow/IntroducingGitFlow.html>. [Consulta: 10 de Mayo de 2018].
- [5] Patrón builder. [https://commons.wikimedia.org/wiki/File:Patron\\_builder.PNG](https://commons.wikimedia.org/wiki/File:Patron_builder.PNG). [Consulta: 10 de Mayo de 2018].
- [6] Patrón dependency inversion. <https://www.pinterest.es/pin/389350330270119451>. [Consulta: 10 de Mayo de 2018].
- [7] Patrón multiton. [https://en.wikipedia.org/wiki/Multiton\\_pattern](https://en.wikipedia.org/wiki/Multiton_pattern). [Consulta: 10 de Mayo de 2018].
- [8] Patrón observer. <http://patronesdiseniophp.blogspot.com.es/p/patron-observer.html>. [Consulta: 10 de Mayo de 2018].
- [9] Patrón singleton. <https://es.wikipedia.org/wiki/Singleton>. [Consulta: 10 de Mayo de 2018].
- [10] Tslint. <https://palantir.github.io/tslint/>. [Consulta: 18 de Mayo de 2018].
- [11] Mike Cohn. *User stories applied: For agile software development*. Addison-Wesley, 2004.
- [12] Mike Cohn. *Succeeding with agile: software development using Scrum*. Addison-Wesley, 2010.



## Anexo A

# Descripción de la metodología *Scrum* para proyectos de software

Basándonos principalmente en la página (*Scrum*) [3] pasamos a describir someramente las principales características de *Scrum*.

*Scrum* es un modelo de desarrollo ágil caracterizado por:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o en cascada.

### A.1. *Sprints*

El desarrollo incremental consiste en dividir el proyecto en bloques temporales (en nuestro caso de 2 semanas) llamados iteraciones o *Sprints*.

El *Sprint* se define como *periodo de tiempo en el cual se produce un Incremento del producto*

Después de cada *Sprint* se consigue un producto funcional llamado *Incremento*. Con ello podemos conseguir *feedback* por parte del cliente y calcular el porcentaje del proyecto desarrollado. En cada iteración se repite un proceso similar al que se realizaría en un proyecto tradicional.

## A.2. Requisitos funcionales

Una de las mayores ventajas que nos ofrece el uso de tecnologías ágiles, y en concreto *Scrum*, es la gestión de los requisitos funcionales del nuevo sistema.

En un sistema tradicional los requisitos se especifican completamente al inicio del proyecto. Cualquier modificación de los mismos a posteriori puede tener un elevado coste de adaptación del desarrollo al cambio.

*Scrum* realiza una gestión de los requisitos funcionales mucho más flexible. Al inicio del proyecto se define la denominada Pila del producto inicial (Initial Product Backlog). Por su nombre ya se intuye que la Pila del producto normalmente sufrirá cambios importantes.

La pila del producto es la lista de los requisitos funcionales que debe cumplir la aplicación. Está en continua evolución y abierta a todos los implicados en el proyecto (desarrolladores, clientes, propietarios del producto,..). La pila del producto es una lista priorizada en función del valor que aportan para el cliente final. También se valora el esfuerzo necesario para cumplir cada requisito en unidades denominadas puntos de historia. Su nombre se debe a que habitualmente los requisitos están expresados en forma de Historia de usuario.

Una vez elaborada la pila del producto estimada y priorizada se planifica un *Spring* creando la pila del *Spring*, que es un subconjunto de la Pila del producto. De cada uno de los requisitos de la Pila del *Spring* se descompone en Tareas estimadas en horas de trabajo.

Al finalizar el *Spring* se obtiene un Incremento del producto, que aporta un valor añadido para el cliente final. Es decir, tendremos un producto con más funcionalidad que la versión producida tras el anterior *Sprint*.

Al concluir el *Sprint* la pila del producto se modifica eliminando los requisitos completados, modificando o eliminando otros y añadiendo otros requisitos nuevos o tareas a desarrollar.

## A.3. Reuniones

- **Planificación del *Sprint*** En esta reunión, los miembros del equipo de trabajo, el *Scrum Master* y el *Product Owner* dividen el proyecto en etapas y tareas. Debe haber un responsable por cada tarea, el cual se define en función de su capacidad y una lógica estimación del esfuerzo. Esta reunión puede durar entre 4 y 8 horas.
- ***Daily Scrum*** Esta reunión tiene lugar cada día, es informal (de pie) y no dura más de 15 minutos. En ella, cada uno de los miembros del equipo de trabajo cuenta brevemente qué hizo en la jornada pasada, lo que hará ahora y los obstáculos que ha ido encontrando. No es una reunión descriptiva ni expositiva; se trata de ir al grano. El *Scrum Master* toma nota de ello.
- **Revisión del *Sprint*** Aunque en algunos casos se plantea como una reunión de cierre, lo cierto es que en ella el equipo de trabajo enseña el *Incremento* del producto. El *Product Owner* acude de forma activa: es el que se encarga de validar o no los resultados. En el



caso del *Scrum Master*, debe tomar nota de aquellos aspectos por mejorar en caso de que los haya. A esta reunión también pueden asistir personas ajenas al proyecto en calidad de observadores.

- **Retrospectiva** Cuando los proyectos culminan, el método *Scrum* plantea una última reunión, la de retrospectiva. En ella, el equipo de trabajo y el *Scrum Master* analizan lo que ha sido el proceso e identifican y evalúan problemas concretos. A veces se trata de problemas menores; otras, en cambio, de asuntos de fondo que tienen que ver con las dinámicas del grupo. Por ello se recomienda que a esta reunión no acudan personas ajenas al equipo de trabajo. Se estima que puede durar 2 o 4 horas y que se haga inmediatamente después de la demo del *Sprint*.



## Anexo B

### Datos en formato JSON para los tests de aceptación

```
{ "idContrato": "000002", "idCliente": "0000010040", "descripcion": "RSU", "fInicio": "694220400000", "fFinalizacion": "1009753200000", "prorrogas": "31/12/2006 Se podrá acordar prorroga 5 años con 3 mese de antelación", "numContenedores": null, "descripcionMejoras": "1.- Limpieza especial con barredora y brigada 2 veces año\r\n2.- Limpieza de Imbornales 3 jornadas año\r\n3.- Limpieza mercado semanal\r\n", "observaciones": null, "idTipoContrato": null, "importeAdjudicacion": null, "revisionesEconomicas": null, "fProrrogalInicial": null, "mejoras": [],
```

"temporadas":

[

```
{ "idContrato": "000002", "idTemporada": "02", "fInicio": "1157061600000", "fFinalizacion": "1183154400000", "descripcion": "TEMPORADA BAJA", "observaciones": null, "prioridad": 0, "temporadaIni": "01/09", "temporadaFin": "30/06",
```

"expedientesContrato":

[

```
{ "idContrato": "000002", "idTemporada": "02", "idDptoExp": "7008", "numExpediente": "000001", "observaciones": null,
```

"plicas":

[

```
{ "idContrato": "000002", "idTemporada": "02", "idDptoExp": "7008", "numExpediente": "000001", "lineaCat": 1, "fechaInicio": "1188597600000", "fechaFin": "1214776800000", "idCategoriaPlicas": "3010", "unidades": 1.0, "planificacion": "Diaria", "horasUd": 4.0, "planifDescHoras": "horas al día", "diaFijoL": "S", "diaFijoM": "S", "diaFijoX": "S", "diaFijoJ": "S", "diaFijoV": "S", "diaFijoS": "S", "diaFijoD": "N", "precioHora": 11.63, "motivoAmpliacion": null, "observaciones": null,
```

```
"categoriaPlicas": { "idCategoriaPlicas": "3010", "descripcion": "Conductor noche RSU" }
```

},

```
{ "idContrato": "000002", "idTemporada": "02", "idDptoExp": "7008", "numExpediente": "000001", "lineaCat": 5, "fechaInicio": "1188597600000", "fechaFin": "1214776800000", "idCategoriaPlicas": "4057", "unidades": 1.0, "planificacion": "Diaria", "horasUd": 4.0, "planifDescHoras": "horas al día", "diaFijoL": "N", "diaFijoM": "N", "diaFijoX": "N", "diaFijoJ": "N", "diaFijoV": "N", "diaFijoS": "N", "diaFijoD": "S", "precioHora": 10.69, "motivoAmpliacion": null, "observaciones": null,
```

```

"categoriaPlicas":{"idCategoriaPlicas":"4057","descripcion":"Peón
noche RSU"}

},

{"idContrato":"000002","idTemporada":"02","idDptoExp":"7008","nu
mExpediente":"000001","lineaCat":8,"fechaInicio":1220220000000,"fe
chaFin":1246312800000,"idCategoriaPlicas":"4055","unidades":1.0,"pl
anificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al
día","diaFijoL":"S","diaFijoM":"S","diaFijoX":"S","diaFijoJ":"S","diaFijoV
":"S","diaFijoS":"S","diaFijoD":"S","precioHora":15.96,"motivoAmpliacion":null,"observaciones":null,"categoriaPlicas":{"idCategoriaPlicas":"40
55","descripcion":"Camión
trasero"}}, {"idContrato":"000002","idTemporada":"02","idDptoExp":"7
008","numExpediente":"000001","lineaCat":2,"fechaInicio":118859760
0000,"fechaFin":1214776800000,"idCategoriaPlicas":"3010","unidades
":1.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas
al
día","diaFijoL":"N","diaFijoM":"N","diaFijoX":"N","diaFijoJ":"N","diaFijo
V":"N","diaFijoS":"N","diaFijoD":"S","precioHora":11.63,"motivoAmplia
cion":null,"observaciones":null,

"categoriaPlicas":{"idCategoriaPlicas":"3010","descripcion":"Conductor
noche RSU"}

},

{"idContrato":"000002","idTemporada":"02","idDptoExp":"7008","nu
mExpediente":"000001","lineaCat":15,"fechaInicio":1251756000000,"f
echaFin":1277848800000,"idCategoriaPlicas":"4057","unidades":1.0,"p
lanificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al
día","diaFijoL":"N","diaFijoM":"N","diaFijoX":"N","diaFijoJ":"N","diaFijo
V":"N","diaFijoS":"N","diaFijoD":"S","precioHora":10.69,"motivoAmplia
cion":null,"observaciones":null,

"categoriaPlicas":{"idCategoriaPlicas":"4057","descripcion":"Peón
noche
RSU"}}, {"idContrato":"000002","idTemporada":"02","idDptoExp":"700
8","numExpediente":"000001","lineaCat":4,"fechaInicio":11885976000
00,"fechaFin":1214776800000,"idCategoriaPlicas":"4057","unidades":1
.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al
día","diaFijoL":"S","diaFijoM":"S","diaFijoX":"S","diaFijoJ":"S","diaFijoV
":"S","diaFijoS":"S","diaFijoD":"N","precioHora":10.69,"motivoAmplia
cion":null,"observaciones":null,

"categoriaPlicas":{"idCategoriaPlicas":"4057","descripcion":"Peón
noche RSU"}

```

```

},

{"idContrato":"000002","idTemporada":"02","idDptoExp":"7008","numExpediente":"000001","lineaCat":6,"fechaInicio":122022000000,"fechaFin":1246312800000,"idCategoriaPlicas":"3010","unidades":1.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al día","diaFijoL":"S","diaFijoM":"S","diaFijoX":"S","diaFijoJ":"S","diaFijoV":"S","diaFijoS":"S","diaFijoD":"N","precioHora":11.63,"motivoAmpliacion":null,"observaciones":null,

"categoriaPlicas":{"idCategoriaPlicas":"3010","descripcion":"Conductor noche RSU"}

},

{"idContrato":"000002","idTemporada":"02","idDptoExp":"7008","numExpediente":"000001","lineaCat":7,"fechaInicio":122022000000,"fechaFin":1246312800000,"idCategoriaPlicas":"3010","unidades":1.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al día","diaFijoL":"N","diaFijoM":"N","diaFijoX":"N","diaFijoJ":"N","diaFijoV":"N","diaFijoS":"N","diaFijoD":"S","precioHora":11.63,"motivoAmpliacion":null,"observaciones":null,

"categoriaPlicas":{"idCategoriaPlicas":"3010","descripcion":"Conductor noche RSU"}

},

{"idContrato":"000002","idTemporada":"02","idDptoExp":"7008","numExpediente":"000001","lineaCat":3,"fechaInicio":1188597600000,"fechaFin":1214776800000,"idCategoriaPlicas":"4055","unidades":1.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al día","diaFijoL":"S","diaFijoM":"S","diaFijoX":"S","diaFijoJ":"S","diaFijoV":"S","diaFijoS":"S","diaFijoD":"S","precioHora":15.96,"motivoAmpliacion":null,"observaciones":null,

"categoriaPlicas":{"idCategoriaPlicas":"4055","descripcion":"Camión trasero"}

},

{"idContrato":"000002","idTemporada":"02","idDptoExp":"7008","numExpediente":"000001","lineaCat":9,"fechaInicio":122022000000,"fechaFin":1246312800000,"idCategoriaPlicas":"4057","unidades":1.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al día","diaFijoL":"S","diaFijoM":"S","diaFijoX":"S","diaFijoJ":"S","diaFijoV":"S","diaFijoS":"S","diaFijoD":"N","precioHora":10.69,"motivoAmpliacion":null,"observaciones":null,"categoriaPlicas":{"idCategoriaPlicas":"40

```

57","descripcion":"Peón noche  
RSU"}},{"idContrato":"000002","idTemporada":"02","idDptoExp":"7008","numExpediente":"000001","lineaCat":11,"fechaInicio":1251756000000,"fechaFin":1277848800000,"idCategoriaPlicas":"3010","unidades":1.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al día","diaFijoL":"S","diaFijoM":"S","diaFijoX":"S","diaFijoJ":"S","diaFijoV":"S","diaFijoS":"S","diaFijoD":"N","precioHora":11.63,"motivoAmpliacion":null,"observaciones":null,

"categoriaPlicas":{"idCategoriaPlicas":"3010","descripcion":"Conductor noche RSU"}

},

{"idContrato":"000002","idTemporada":"02","idDptoExp":"7008","numExpediente":"000001","lineaCat":12,"fechaInicio":1251756000000,"fechaFin":1277848800000,"idCategoriaPlicas":"3010","unidades":1.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al día","diaFijoL":"N","diaFijoM":"N","diaFijoX":"N","diaFijoJ":"N","diaFijoV":"N","diaFijoS":"N","diaFijoD":"S","precioHora":11.63,"motivoAmpliacion":null,"observaciones":null,

"categoriaPlicas":{"idCategoriaPlicas":"3010","descripcion":"Conductor noche RSU"}

},

{"idContrato":"000002","idTemporada":"02","idDptoExp":"7008","numExpediente":"000001","lineaCat":14,"fechaInicio":1251756000000,"fechaFin":1277848800000,"idCategoriaPlicas":"4057","unidades":1.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al día","diaFijoL":"S","diaFijoM":"S","diaFijoX":"S","diaFijoJ":"S","diaFijoV":"S","diaFijoS":"S","diaFijoD":"N","precioHora":10.69,"motivoAmpliacion":null,"observaciones":null,"categoriaPlicas":{"idCategoriaPlicas":"4057","descripcion":"Peón noche RSU"}

},

{"idContrato":"000002","idTemporada":"02","idDptoExp":"7008","numExpediente":"000001","lineaCat":10,"fechaInicio":1220220000000,"fechaFin":1246312800000,"idCategoriaPlicas":"4057","unidades":1.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al día","diaFijoL":"N","diaFijoM":"N","diaFijoX":"N","diaFijoJ":"N","diaFijoV":"N","diaFijoS":"N","diaFijoD":"S","precioHora":10.69,"motivoAmpliacion":null,"observaciones":null,"categoriaPlicas":{"idCategoriaPlicas":"4057","descripcion":"Peón noche RSU"}

},

```

        {"idContrato":"000002","idTemporada":"02","idDptoExp":"7008","numExpediente":"000001","lineaCat":13,"fechaInicio":1251756000000,"fechaFin":1277848800000,"idCategoriaPlicas":"4055","unidades":1.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al día","diaFijoL":"S","diaFijoM":"S","diaFijoX":"S","diaFijoJ":"S","diaFijoV":"S","diaFijoS":"S","diaFijoD":"S","precioHora":15.96,"motivoAmpliacion":null,"observaciones":null,

        "categoriaPlicas":{"idCategoriaPlicas":"4055","descripcion":"Camión trasero"}}

    ],

    "expediente":{"idDepartamento":"7008","numExpediente":"000001","idCentroCoste":"0300010097","descripcion":"Alquilerias RSU"}

    , "departamento":{"idDepartamento":"7008","nombre":"AREA DE CASTELLÓN"}

}

]

},

{"idContrato":"000002","idTemporada":"01","fInicio":1183240800000,"fFinalizacion":1188511200000,"descripcion":"TEMPORADA ALTA","observaciones":null,"prioridad":0,"temporadaIni":"01/07","temporadaFin":"31/08",

"expedientesContrato":

[

{"idContrato":"000002","idTemporada":"01","idDptoExp":"7008","numExpediente":"000001","observaciones":null,"plicas":[{"idContrato":"000002","idTemporada":"01","idDptoExp":"7008","numExpediente":"000001","lineaCat":9,"fechaInicio":1246399200000,"fechaFin":1251669600000,"idCategoriaPlicas":"4057","unidades":1.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al día","diaFijoL":"S","diaFijoM":"S","diaFijoX":"S","diaFijoJ":"S","diaFijoV":"S","diaFijoS":"S","diaFijoD":"N","precioHora":10.69,"motivoAmpliacion":null,"observaciones":null,

"categoriaPlicas":{"idCategoriaPlicas":"4057","descripcion":"Peón noche RSU"}}

],

{"idContrato":"000002","idTemporada":"01","idDptoExp":"7008","numExpediente":"000001","lineaCat":13,"fechaInicio":1277935200000,"fechaFin":1283205600000,"idCat

```



egoriaPlicas": "4055", "unidades": 1.0, "planificacion": "Diaria", "horasUd": 4.0, "planifDesc Horas": "horas al día", "diaFijoL": "S", "diaFijoM": "S", "diaFijoX": "S", "diaFijoJ": "S", "diaFijoV": "S", "diaFijoS": "S", "diaFijoD": "S", "precioHora": 15.96, "motivoAmpliacion": "Ampliación los domingos por petición del Ayto", "observaciones": null, "categoriaPlicas": {"idCategoriaPlicas": "4055", "descripcion": "Camión trasero"}

},

{"idContrato": "000002", "idTemporada": "01", "idDptoExp": "7008", "numExpediente": "000001", "lineaCat": 14, "fechaInicio": 1277935200000, "fechaFin": 1283205600000, "idCategoriaPlicas": "4057", "unidades": 1.0, "planificacion": "Diaria", "horasUd": 4.0, "planifDesc Horas": "horas al día", "diaFijoL": "S", "diaFijoM": "S", "diaFijoX": "S", "diaFijoJ": "S", "diaFijoV": "S", "diaFijoS": "S", "diaFijoD": "N", "precioHora": 10.69, "motivoAmpliacion": null, "observaciones": null, "categoriaPlicas": {"idCategoriaPlicas": "4057", "descripcion": "Peón noche RSU"}}

},

{"idContrato": "000002", "idTemporada": "01", "idDptoExp": "7008", "numExpediente": "000001", "lineaCat": 5, "fechaInicio": 1214863200000, "fechaFin": 1220133600000, "idCategoriaPlicas": "4057", "unidades": 1.0, "planificacion": "Diaria", "horasUd": 4.0, "planifDesc Horas": "horas al día", "diaFijoL": "N", "diaFijoM": "N", "diaFijoX": "N", "diaFijoJ": "N", "diaFijoV": "N", "diaFijoS": "N", "diaFijoD": "S", "precioHora": 10.69, "motivoAmpliacion": null, "observaciones": null, "categoriaPlicas": {"idCategoriaPlicas": "4057", "descripcion": "Peón noche RSU"}}

},

{"idContrato": "000002", "idTemporada": "01", "idDptoExp": "7008", "numExpediente": "000001", "lineaCat": 15, "fechaInicio": 1277935200000, "fechaFin": 1283205600000, "idCategoriaPlicas": "4057", "unidades": 1.0, "planificacion": "Diaria", "horasUd": 4.0, "planifDesc Horas": "horas al día", "diaFijoL": "N", "diaFijoM": "N", "diaFijoX": "N", "diaFijoJ": "N", "diaFijoV": "N", "diaFijoS": "N", "diaFijoD": "S", "precioHora": 10.69, "motivoAmpliacion": "Ampliación por petición del Ayto", "observaciones": null,

"categoriaPlicas": {"idCategoriaPlicas": "4057", "descripcion": "Peón noche RSU"}}

},

{"idContrato": "000002", "idTemporada": "01", "idDptoExp": "7008", "numExpediente": "000001", "lineaCat": 11, "fechaInicio": 1277935200000, "fechaFin": 1283205600000, "idCategoriaPlicas": "3010", "unidades": 1.0, "planificacion": "Diaria", "horasUd": 4.0, "planifDesc Horas": "horas al día", "diaFijoL": "S", "diaFijoM": "S", "diaFijoX": "S", "diaFijoJ": "S", "diaFijoV": "S", "diaFijoS": "S", "diaFijoD": "N", "precioHora": 11.63, "motivoAmpliacion": null, "observaciones": null,

```

"categoriaPlicas":{"idCategoriaPlicas":"3010","descripcion":"Conductor noche RSU"}
},

{"idContrato":"000002","idTemporada":"01","idDptoExp":"7008","numExpediente":"000001","lineaCat":12,"fechaInicio":1277935200000,"fechaFin":1283205600000,"idCategoriaPlicas":"3010","unidades":1.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al día","diaFijoL":"N","diaFijoM":"N","diaFijoX":"N","diaFijoJ":"N","diaFijoV":"N","diaFijoS":"N","diaFijoD":"S","precioHora":11.63,"motivoAmpliacion":"Ampliación por petición del Ayto","observaciones":null,

"categoriaPlicas":{"idCategoriaPlicas":"3010","descripcion":"Conductor noche RSU"}
},

{"idContrato":"000002","idTemporada":"01","idDptoExp":"7008","numExpediente":"000001","lineaCat":10,"fechaInicio":1246399200000,"fechaFin":1251669600000,"idCategoriaPlicas":"4057","unidades":1.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al día","diaFijoL":"N","diaFijoM":"N","diaFijoX":"N","diaFijoJ":"N","diaFijoV":"N","diaFijoS":"N","diaFijoD":"S","precioHora":10.69,"motivoAmpliacion":"Ampliación por petición del Ayto","observaciones":null,"categoriaPlicas":{"idCategoriaPlicas":"4057","descripcion":"Peón noche RSU"}
},

{"idContrato":"000002","idTemporada":"01","idDptoExp":"7008","numExpediente":"000001","lineaCat":4,"fechaInicio":1214863200000,"fechaFin":1220133600000,"idCategoriaPlicas":"4057","unidades":1.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al día","diaFijoL":"S","diaFijoM":"S","diaFijoX":"S","diaFijoJ":"S","diaFijoV":"S","diaFijoS":"S","diaFijoD":"N","precioHora":10.69,"motivoAmpliacion":null,"observaciones":null,

"categoriaPlicas":{"idCategoriaPlicas":"4057","descripcion":"Peón noche RSU"}
},

{"idContrato":"000002","idTemporada":"01","idDptoExp":"7008","numExpediente":"000001","lineaCat":2,"fechaInicio":1214863200000,"fechaFin":1220133600000,"idCategoriaPlicas":"3010","unidades":1.0,"planificacion":"Diaria","horasUd":4.0,"planifDescHoras":"horas al día","diaFijoL":"N","diaFijoM":"N","diaFijoX":"N","diaFijoJ":"N","diaFijoV":"N","diaFijoS":"N","diaFijoD":"S","precioHora":11.63,"motivoAmpliacion":null,"observaciones":null,

"categoriaPlicas":{"idCategoriaPlicas":"3010","descripcion":"Conductor noche RSU"}
},

```

```
{ "idContrato": "000002", "idTemporada": "01", "idDptoExp": "7008", "numExpediente": "000001", "lineaCat": 8, "fechaInicio": 1246399200000, "fechaFin": 1251669600000, "idCategoriaPlicas": "4055", "unidades": 1.0, "planificacion": "Diaria", "horasUd": 4.0, "planifDescHoras": "horas al día", "diaFijoL": "S", "diaFijoM": "S", "diaFijoX": "S", "diaFijoJ": "S", "diaFijoV": "S", "diaFijoS": "S", "diaFijoD": "S", "precioHora": 15.96, "motivoAmpliacion": "Ampliación los domingos por petición del Ayto", "observaciones": null,
```

```
"categoriaPlicas": { "idCategoriaPlicas": "4055", "descripcion": "Camión trasero" }
```

```
},
```

```
{ "idContrato": "000002", "idTemporada": "01", "idDptoExp": "7008", "numExpediente": "000001", "lineaCat": 6, "fechaInicio": 1246399200000, "fechaFin": 1251669600000, "idCategoriaPlicas": "3010", "unidades": 1.0, "planificacion": "Diaria", "horasUd": 4.0, "planifDescHoras": "horas al día", "diaFijoL": "S", "diaFijoM": "S", "diaFijoX": "S", "diaFijoJ": "S", "diaFijoV": "S", "diaFijoS": "S", "diaFijoD": "N", "precioHora": 11.63, "motivoAmpliacion": null, "observaciones": null,
```

```
"categoriaPlicas": { "idCategoriaPlicas": "3010", "descripcion": "Conductor noche RSU" }
```

```
},
```

```
{ "idContrato": "000002", "idTemporada": "01", "idDptoExp": "7008", "numExpediente": "000001", "lineaCat": 7, "fechaInicio": 1246399200000, "fechaFin": 1251669600000, "idCategoriaPlicas": "3010", "unidades": 1.0, "planificacion": "Diaria", "horasUd": 4.0, "planifDescHoras": "horas al día", "diaFijoL": "N", "diaFijoM": "N", "diaFijoX": "N", "diaFijoJ": "N", "diaFijoV": "N", "diaFijoS": "N", "diaFijoD": "S", "precioHora": 11.63, "motivoAmpliacion": "Ampliación por petición del Ayto", "observaciones": null,
```

```
"categoriaPlicas": { "idCategoriaPlicas": "3010", "descripcion": "Conductor noche RSU" }
```

```
},
```

```
{ "idContrato": "000002", "idTemporada": "01", "idDptoExp": "7008", "numExpediente": "000001", "lineaCat": 1, "fechaInicio": 1214863200000, "fechaFin": 1220133600000, "idCategoriaPlicas": "3010", "unidades": 1.0, "planificacion": "Diaria", "horasUd": 4.0, "planifDescHoras": "horas al día", "diaFijoL": "S", "diaFijoM": "S", "diaFijoX": "S", "diaFijoJ": "S", "diaFijoV": "S", "diaFijoS": "S", "diaFijoD": "N", "precioHora": 11.63, "motivoAmpliacion": null, "observaciones": null,
```

```
"categoriaPlicas": { "idCategoriaPlicas": "3010", "descripcion": "Conductor noche RSU" }
```

```
},
```

```
{ "idContrato": "000002", "idTemporada": "01", "idDptoExp": "7008", "numExpediente": "000001", "lineaCat": 3, "fechaInicio": 1214863200000, "fechaFin": 1220133600000, "idCategoriaPlicas": "4055", "unidades": 1.0, "planificacion": "Diaria", "horasUd": 4.0, "planifDescHoras": "horas al día", "diaFijoL": "S", "diaFijoM": "S", "diaFijoX": "S", "diaFijoJ": "S", "diaFijoV": "S", "diaFijoS": "S", "diaFijoD": "N", "precioHora": 11.63, "motivoAmpliacion": null, "observaciones": null,
```

```

oras":"horas al
día","diaFijoL":"S","diaFijoM":"S","diaFijoX":"S","diaFijoJ":"S","diaFijoV":"S","diaFijoS":
"S","diaFijoD":"S","precioHora":15.96,"motivoAmpliacion":null,"observaciones":null,"c
ategoriaPlicas":{"idCategoriaPlicas":"4055","descripcion":"Camión trasero"}

},

"expediente":{"idDepartamento":"7008","numExpediente":"000001","idCentroCoste":
"0300010097","descripcion":"Alquilerias
RSU"},"departamento":{"idDepartamento":"7008","nombre":"AREA DE
CASTELLÓN}}}},{"tipoContrato":null,"cliente":{"idCliente":"0000010040","razonSocial"
:"AYUNTAMIENTO ALQUERIAS DEL NIÑO PER","nombreComercial":"AYTO.ALQUE"}

}

```

## Anexo C

### Sentencias de la base de datos H2 para tests

```
SET REFERENTIAL_INTEGRITY FALSE;
```

```
truncate table ContratoObTipo;  
truncate table ContratoOb ;  
truncate table ContratoObMejora ;  
truncate table ContratoObTemporada ;  
truncate table ContratoObTemporadaExp ;  
truncate table ContratoObTemporadaExpcatPlicas ;  
truncate table CategoriaPlicas ;  
truncate table Expediente ;  
truncate table Cliente ;  
truncate table Departamento ;
```

```
truncate table INFListadoCoLIN;  
truncate table INFClaseCoste;  
truncate table Expediente;
```

```
truncate table ParteTrabajoCab;  
truncate table ParteTrabajoHEmp;  
truncate table ParteTrabajoHVeh;  
truncate table Vehiculo;
```

```
SET REFERENTIAL_INTEGRITY TRUE;
```

```
-- Cliente
```

```
INSERT INTO `Cliente`(`IdCliente`, `RazonSocial`, `NombreCom`)  
VALUES ( '0001', 'Cliente 0001', 'Cliente 1');
```

```
-- Cliente
```

```
INSERT INTO `Departamento`(`IdDepartamento`, `Nombre`)  
VALUES ( '7008', 'Departamento 7008');
```

```
-- ContratoObTipo
```

```
INSERT INTO `ContratoObTipo`(`IdTipoContratoOb`, `Descripcion`)  
VALUES ( '1', 'LV');
```

```
-- Contratos obligación
```

```
INSERT INTO `ContratoOb`(  
    `IdContratoOb`, `IdCliente`, `Descripcion`, `FInicio`, `FFinalizacion`,  
    `Prorrogas`, `NumContenedores`, `Mejoras`, `Observaciones`, `IdTipoContratoOb`,  
    `ImporteAdjudicacion`, `RevisionesEconomicas`, `FProrrogaInicial`
```

```

)
VALUES (
    '1', '0001', 'Contrato obligación de ejemplo', {ts '2018-01-01 00:00:00.00'}, {ts '2018-12-31
00:00:00.00'},
    'Sin prorrogas', 1, 'Sin mejoras', 'Observaciones de prueba', 1,
    1000.0, 'Sin revisiones', {ts '2019-01-01 00:00:00.00'}
);

-- Mejoras de contrato obligacion

INSERT INTO `ContratoObMejora` (
    `IdContratoOb`, `IdMejora`, `Descripcion`, `TotalAnyo`, `TotalContrato`, `CosteReal`,
    `Puntuacion`, `realizada`)
VALUES (
    '1', 1, 'Descripción de la mejora', '1000.0', '5000.0', '500.0', 1, '1'
);

-- Temporadas de contrato obligacion

INSERT INTO `ContratoObTemporada` (`IdContratoOb`, `IdTemporada`, `Finicio`, `FFinalizacion`,
`Descripcion`,
    `Observaciones`, `Prioridad`, `TemporadaIni`, `TemporadaFin`)
VALUES (
    '1', '1', {ts '2018-01-01 00:00:00.00'}, {ts '2018-12-31 00:00:00.00'}, 'Temporada alta',
    'Observaciones de Temporada', 1, '01/01', '31/12'
);

-- Expedientes

INSERT INTO `Expediente` (`IdDepartamento`, `NumExpediente`, `Descripcion`)
VALUES (
    '7008', '000093', 'Benicasim RSU'
);

-- Expedientes de contrato obligacion

INSERT INTO `ContratoObTemporadaExp` (`IdContratoOb`, `IdTemporada`, `IdDptoExp`,
`NumExpediente`,
    `Observaciones`)
VALUES (
    '1', '1', '7008', '000093', 'Observaciones de Expediente'
);

-- CategoriaPlicas

```

```

INSERT INTO `CategoriaPlicas`(
    `IdcategoriaPlicas`, `Descripcion`)
VALUES (
    '3010', 'Conductor noche RSU'
);

-- CategoriaPlicas de contrato obligacion

INSERT INTO `ContratoObTemporadaExpcatPlicas`(
    `IdContratoOb`, `IdTemporada`, `IdDptoExp`, `NumExpediente`, `LineaCat`, `FechaInicio`,
    `FechaFin`,
    `IdCategoriaPlicas`, `Unidades`, `Planificacion`, `HorasUd`, `PlanifDescHoras`,
    `DiaFijoL`, `DiaFijoM`, `DiaFijoX`, `DiaFijoJ`, `DiaFijoV`, `DiaFijoS`, `DiaFijoD`,
    `PrecioHora`, `MotivoAmpliacion`, `Observaciones`)
VALUES (
    '1', '1', '7008', '000093', '1', {ts '2018-01-01 00:00:00.00'}, {ts '2018-12-31 00:00:00.00'},
    '3010', 1.00, 'Diaria', 4.00, 'horas al dia',
    'S', 'S', 'S', 'S', 'S', 'S', 'S',
    11.6300, 'Ampliación por petición del Ayto', 'Observaciones de Plicas'
);

-- INFClaseCoste

INSERT INTO `INFClaseCoste`(
    `IdClaseCoste`, `Descripcion`
)
VALUES (
    '0060320000', 'Trabajos realizados por Administración'
);

-- Expediente

INSERT INTO `Expediente`(
    `IdDepartamento`, `NumExpediente`, `IdCentroCoste`
)
VALUES (
    '7008', '001', '0000110010'
);

-- INFListadoCoLIN

INSERT INTO `INFListadoCoLIN`(
    `IdDocumento`, `IdLinea`, `Importe`, `IdCeCo`, `IdClaseCoste`, `Fecha`
)
VALUES (

```



```

'010000000002', '001', 10.5, '0000110010', '0060320000', '20181230'
);

INSERT INTO `INFListadoCoLIN` (
  `IdDocumento`, `IdLinea`, `Importe`, `IdCeCo`, `IdClaseCoste`, `Fecha`
)
VALUES (
  '010000000003', '001', 10.5, '0000110010', '0060320000', '20181231'
);

-- ParteTrabajoCab

INSERT INTO `ParteTrabajoCab` (
  `IdDepartamento`, `IdAnyoParte`, `IdParte`, `NumExpedienteHoras`, `IdDptoExpHoras`,
  `FFintrabajos`
)
VALUES (
  '7008', '2018', '000043', '001', '7008', {ts '2018-01-01 00:00:00.00'}
);

INSERT INTO `ParteTrabajoCab` (
  `IdDepartamento`, `IdAnyoParte`, `IdParte`, `NumExpedienteHoras`, `IdDptoExpHoras`,
  `FFintrabajos`
)
VALUES (
  '7008', '2018', '000044', '001', '7008', {ts '2018-01-02 00:00:00.00'}
);

-- ParteTrabajoHEmp

INSERT INTO `ParteTrabajoHEmp` (
  `IdDepartamento`, `IdAnyoParte`, `IdParte`, `LineaHEmp`, `IdCategoriaPlicas`,
  `HorasNormales`,
  `HorasExtras`, `HorasExtraCompensacion`
)
VALUES (
  '7008', '2018', '000043', 1, '3010', '10.0', '10.0', '10.0'
);

INSERT INTO `ParteTrabajoHEmp` (
  `IdDepartamento`, `IdAnyoParte`, `IdParte`, `LineaHEmp`, `IdCategoriaPlicas`,
  `HorasNormales`,
  `HorasExtras`, `HorasExtraCompensacion`
)
VALUES (

```

```
'7008', '2018', '000044', 1, '3010', '10.0', '10.0', '10.0'  
);
```

-- Vehiculo

```
INSERT INTO `Vehiculo`(  
    `IdVehiculo`, `IdCategoriaPlicas`  
)  
VALUES (  
    'CS8096AH', '3010'  
);
```

-- ParteTrabajoHVeh

```
INSERT INTO `ParteTrabajoHVeh`(  
    `IdDepartamento`, `IdAnyoParte`, `IdParte`, `LineaHVeh`, `IdVehiculo`, `Horas`  
)  
VALUES (  
    '7008', '2018', '000043', 1, 'CS8096AH', '10.0'  
);
```

```
INSERT INTO `ParteTrabajoHVeh`(  
    `IdDepartamento`, `IdAnyoParte`, `IdParte`, `LineaHVeh`, `IdVehiculo`, `Horas`  
)  
VALUES (  
    '7008', '2018', '000044', 1, 'CS8096AH', '10.0'  
);
```